IPSO FACTO

Issue #9
December, 1978

(A publication of the Association of Computer Experimenters)

TABLE OF CONTENTS PAGE

ACE Executive & Meeting Schedule..... 2
Editor's Remarks..... 3
Some Thoughts on the CALL and RETURN Technique..... 4
COSMAC ELF Reference..... 6
Automatic Program Counter Stepper..... 6
Notes on Netronics' TINY BASIC..... 7
Some ELF II Enhancements..... 8
Some Simple 1802 Programs..... 9
Programming the DM8577 (32X8) PROM..... 10
PING-PONG..... 13
ACE Tutorial..... 18
Fix for 1861 Video Display Jitter..... 18
Radio Shack Keyboard Enhancement..... 18
New Basic for COSMAC 1802 Available..... 19
Netronics Monitor Disassembled..... 20
Partial Display Routines for ELF Systems..... 22
Interesting Computer Articles..... 22
ERRATA..... 24
Running IPSO FACTO Cassette Software on the ELF.... 24
Netronics' "Giant Board" Notes..... 25
Letters to the Editor..... 25
For Sale..... 28
Letters of Contact..... 29
STUDIO II Conversion Kit..... 30
Tic-Tac-Toe with the 1802..... 31
Key In Loader for ELF II Format Files..... 34
LIFE for an ELF..... 36
Nominations and Elections..... 38
Minutes of ACE Meetings..... 39
IEEE Microprocessor Course Notice..... 41
Renewal and Application Forms..... 42

Editor: Bernie Murphy
Invaluable Assistants: Wayne Bowdish, Tom Crawford, Vic Sydiuk, Diane York, and all contributors to this issue.

Information furnished by IPSO FACTO is believed to be accurate and reliable. However, no responsibility is assumed by IPSO FACTO or the Association of Computer Experimenters for its use; nor for any infringements of patents or other rights of third parties which may result from its use.

Send Newsletter correspondence to: Bernie Murphy
102 McCrany Street,
Oakville, Ontario
Canada L6H 1H6

ACE EXECUTIVE COMMITTEE

In accordance with the Constitution, the 1978-79 Executive Committee approved at the Annual General Meeting is:

President	KEN BEVIS	220 Cherry Post Drive, Missisauga, Ontario, L5A 1H9 (277-2495)
Past President	TOM CRAWFORD	50 Brentwood Drive, Stoney Creek, Ontario, L8G 2W8 (662-3603
Secretary/ Treasurer	GEORGE YORK	60 Chester Road, Stoney Creek, Ontario, L8E 1Y2 (664-5264)
Newsletter Editor	BERNIE MURPHY	102 McCrany Street, Oakville, Ontario, L6H 1H6 (845-1630
Program Co-ordinator	BERT DEKAT	P.O. Box 137, Lynden, Ontario LOR 1TO (647-3931)
Training Co-ordinator	ROD DORE	660 Oxford Road, Unit 32, Burlington, Ontario, L7N 3M1 (681-2456)
Hardware Co-ordinator	FRED FEAVER	105 Townsend Avenue, Burlington, Ontario, L7T 1Y8 (637-2513)
Membership Co-ordinator	WAYNE BOWDISH (Temporary)	149 East 33rd Street, Hamilton, Ontario, L8V 3T5 (388-7116)
Newsletter Publishing Committee	DENNIS MILDON	44 Wildewood Avenue, Hamilton, Ontario, L8T 1X3 (385-0798)
	JOHN HANSON	955 Harvey Place, Burlington, Ontario, L7T 3E9 (637-1076)

There are still some positions to be filled eg. software and membership co-ordinators. Any volunteers for these positions or to help with any other activity should contact Ken Bevis.

ACE CLUB MEETING SCHEDULE

Unless notice to the contrary, the following is the meeting schedule until the end of May. The executive is trying to arrange tours for the April 10 and May 22 meetings. Details will be published in the next newsletter. The meeting scheduled for March 13 is to have a flea market so start collecting up your surplus.

DATE	TUTORIAL/MEETING	TUTORIAL
Jan 9	7:00/8:00	
23		7:30
Feb 13	7:00/8:00	
27		7:30
Mar 13	7:00/8:00 (FLEA MARKET)	
27		7:30
Apr 10	8:00 (TOUR)	
24		7:30
May 8	7:00/8:00 (ANNUAL GENERAL MEETING)	
22	7:30 (TOUR)	

ALSO--unless notice to the contrary, all meetings will be held in the Stelco Wilcox St. Auditorium.

EDITOR'S REMARKS

I hope that you have recovered from the New Year's festivities by now and have resolved to use your 1802 at least once in 1979.

From what I have been reading in the trade journals, VLSI or Very Large Scale Integration is going to have a drastic effect on what we will be doing 2 or 3 years from now. With 1 million bit memories and 500 nanosecond 16 and 32 bit microprocessors coming up in the future, we must keep in step with technology or we will be "snowed under".

NEW DIGITAL EXPERIMENTER'S LICENSE

Canada's Telecommunications Regulatory Service has given the green light to allow Canadian radio amateurs to experiment with "packet radio". Holders of the new Digital Amateur Radio Operator's Certificate can use all modes available to other classes above 144.0 MHz. The code requirement has been dropped but if the Digital amateur later completed the code requirement, he will be given the "Advanced" Certificate.

If you are into this facet of amateur radio, please tell us what is going on. (What protocol are you using? what code sets?, what microprocessor(s) are people using?, etc.)

PROJECTS

I was investigating the Fuel Save Thermostats that allow you to automatically lower the temperature at night and raise it in the morning. (California readers can skip this section!) Sounds great until you ask the price $80.00!

By using an extra regular thermostat, an 1802, a 2758 EPROM and a few gates one should be able to come up with a super timer system. For example, you could allow for different on/off times for the weekends and even have a "I'm going out for 4 hours" button. By careful and selective shopping you could probably build the above system for less than $75.00.

THE NEWSLETTER

We have been getting several letters lately asking such things as: Where is my newsletter?, how come I have not received my back issues yet, where are they?, etc. Remember...we are not in the publishing business. There is a very small core of devoted people who do all the work as VOLUNTEERS (ie. no pay, no travel expenses, no free meals).

Please be patient! The mail is very slow--it took 3 weeks for Issue #8 to travel 10 miles, via the mail system!

Please keep submitting your articles and ideas. We have, according to many people, one of the best microprocessor newsletters in the world. Let's keep it that way!

SOME THOUGHTS ON THE CALL AND RETURN TECHNIQUE (FOR RELOCATABLE PROGRAMS)

Volker Raab
Ramtenvej 30
DK 8581 Nimtofte
Denmark

Having just ordered my 1802 (a Super Elf) the other day and hoping to receive it before Christmas, I have spent some time thinking about the software.

Despite the examples I have seen in the 1802 manual and by Tom Pitman, I plan to use another call and return technique.

I had just started to write some small pieces of software (to get used to the slightly unusual instruction set) when I immediately ran into a problem. What if I want to add some code at another location, what if I discover a tiny bug? Will I have to change all references in all my programs each time.

I will have none of that.

As of now, I have nearly finished one page of service routines for 16 bit word processing (push, pull, add, sub...). The service routines use one dedicated register:

RWS — word sequence program counter
and know about the following registers used by the calling program:

RPC — calling program program counter

RA — general simulated word accumulator

RW — general scratch register for both program and service routines

RPB — program base page - the memory address of the basepage, the first page of the program, only the high part of the register is actually used, the low part may be used as another scratch register.

EXAMPLE: addition

The addword service routine adds the contents of the word pointed to by the current X-register to the contents of the RA register. The call looks like this:

```
          SEP  RWS  ;call service routines
               F9   ;with function code for addition
```

The call and exit part of the service routines looks like this:

```
SEXIT:    SEP  RPC  ;return to calling program
SENTRY:   LDA  RPC  ;load function code
          PLO  RW   ;use function code as offset to
          GHI  RWS  ;a service routine function table
          PHI  RW   ;
          LDA  RW   ;get table contents for function
          PLO  RWS  ;jump to specific service routine
```

In this particular case, I might do without a function table, but I really am afraid of hidden bugs.

Apart from the obvious functions the service routines contain three other functions called: enter, jump and return. The calling sequence looks like this:

```
          SEP  RWS  ;enter service routines
               FX   ;with function code for enter, jump or exit
               FY   ;and in case of enter or jump,
                    ;another function code for the enter or jump
```

Now, the three functions:

```
ENTER:      GLO  RPC      ;save program  program counter on stack
            STR  R2       ;
            DEC  R2       ;the current content of the X-register
            GHI  RPC      ;is considered as an important parameter
            STR  R2       ;that's why I don't use STXD
            DEC  R2       ;
JUMP:       LDA  RPC      ;get jump function (FY)
            PLO  RPB      ;and corresponding entry in program
            LDA  RPB      ;jump table and finally take the address
            PHI  RPC      ;from the table and put it into the
            LDN  RPB      ;programs program counter
            PLO  RPC      ;
            BR   SEXIT    ;and then exit and jump
EXIT:       INC  R2       ;get the original program counter back
            LDA  R2       ;from the stack
            PHI  RPC      ;
            LDN  R2       ;
            PLO  RPC      ;
            BR   SEXIT    ;
```

Now I have made some implied assumptions about my programs. A program consists of one or more pages, that must be all in memory at the same time. The first page is special, it contains

- an initialization part (I don't know yet what it will look like)
- a program description part (containing the start address of the interpage jump table, to enable a future program loader to change the addresses in the table)
- some instructions of the program or
- some of the variables used by the program and
- the interpage jump and enter table for the program.

Apart from the obvious use of memory for the table, I pay another prize: No program can use more than at most 128 interpage jumps and probably not even half as much. But I hope at that time I might find out, just how one program calls another program, even one that must be loaded, before it can be entered.

I will write my programs with as few interpage jumps as possible and in any case I'll never use a long branch but the interpage jump table. I'll write all programs as if starting at address: 0000. When I load it manually or automatically, I'll just have to change the addresses in the interpage jump table either by hand or by means of a loader program.

I have written all this because I fear I will end up with a powerful but most unusual system, but I hope my ideas will encourage others with similar ideas to get started in the same direction.

COSMAC ELF REFERENCES

M. Skodny
80 Weir St. S.
Hamilton, Ont.
L8K 3A6

I have put together a list of all articles in "Popular Electronics" dealing with RCA-1802 based "Elf" microcomputer.

- Build the Cosmac "Elf". Part 1.
  Construction of a low-cost microcomputer and some simple examples of programming. August 1976.

- Build the Cosmac "Elf". Part 2.
  Some hardware improvements and more programming details. September 1976.

- Build the Cosmac "Elf". Part 3.
  How to expand memory, plus more programs. March 1977.

- Build the PIXIE Graphic Display. Part 4.
  Adding one chip to the Elf provides complete video interface. July 1977.

- How to update a basic "Elf" microcomputer. TTY Interface. Cassette Read/Write. Musical Programs. Frequency Counter. February 1978.

- Expanding the Elf II.
  Boost an Elf from a microcomputer to virtually a personal computer. March 1978.

- Build a Personal Microcomputer for $100.00.
  Video graphic on your TV set, machine language programming, games.
  PE's Electronic Experimenter's Handbook, 1978 Edition.

- A Tic-Tac-Toe Game for your "Elf".
  Use a simple light pen and the programs given here.
  November 1978.

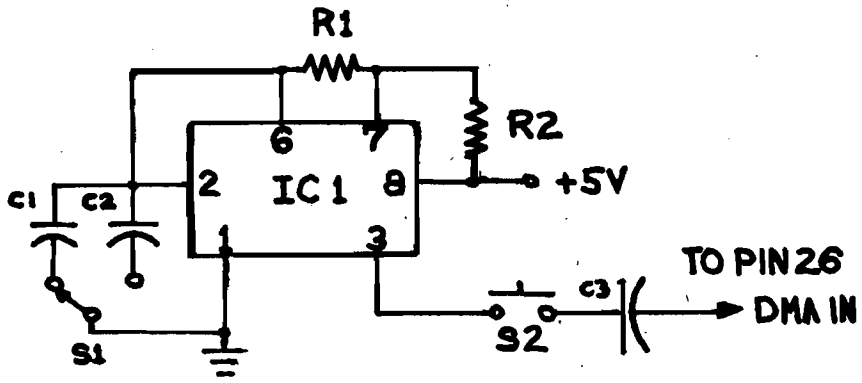
AUTOMATIC PROGRAM COUNTER STEPPER

Christopher Airhart
364 Church St
Stratford, Ont.
N5A 2S1

I have enclosed plans for a circuit to advance the program counter in the Load mode. There are two speed selections.

I have found this circuit especially helpful in de-bugging programs and examining the contents of the memory.

AUTOMATIC PROGRAM COUNTER STEPPER (CONT'D)



PARTS LIST
C1-.02 mfd
C2-.068 mfd
C3-.22 mfd
IC 1-LM555
R1-1.5M ohm
R2-3.3K ohm
S1-SPDT switch
S2-SPST N.O. switch

This circuit is intended to advance the CPU's program counter, R(0), via the DMA-IN channel. By doing so, it is possible to examine the memory contents and modify them.

The heart of the circuit is the LM555, The IC is set up in a stable mode. There are two speed settings, fast or slow, selected by S1. The output is controlled by S2, and debounced by C3. C3 is connected to pin 26 on the TEC-1802 bus.

To use the circuit, switch C/D to C, press RESET, MEMORY PROTECT, and LOAD in the given order. Depressing S2 will advance the program counter. To modify a certain byte, stop at the memory address one byte before the byte, and take off the memory protect (press CE).

For machines other than the TEC-1802, certain changes must be made. The negative pulse width would require shortening (less than 1 machine cycle).

REFERENCES: 1. RCA MPM-201 User's Manual
2. National Linear Data Book
3. Tectron TEC-1802 Assembly Notes

NOTES ON NETRONICS' TINY BASIC

David A. Hersker
Box 802
Conyngham, PA. 18219

Netronics' TINY BASIC is designed to run on the ELF II using their systems monitor/editor to load. TINY requires an absolute minimum of 4.25K memory to run. If not using the Elf II with the "Giant Board", the following hardware specifications must be met. If you are using an ASCII keyboard with parallel data output,

it must go to an input port called by the 6F instruction. Also the data strobe must go to the $\overline{EF3}$ line on the 1802. When using a parallel ASCII keyboard, TINY BASIC acts as a character generator to display the ASCII input/output on the video monitor using the 1861 chip for video out. The 1861 must be turned on by a 69 instruction and turned off by a 61 instruction.

When using a TTY with the TINY BASIC as the input/output, it must interface with the 1802 thru either a 20MA or RS232C Interface with Q being the send and $\overline{EF4}$ being the receive lines. If, as the ELF II does, you use the $\overline{EF4}$ line for your Input switch you will have to install a switch to disable one of the inputs (machine language/ASCII). All this information comes from the "Giant Board" Assembly Instructions but NOT the TINY BASIC Users Manual. I am under the impression that the Netronics systems monitor/editor tape read/write routines approximate the Kansas City Standard (POPULAR ELECTRONICS, February 1978 lists tape read/write routines which they imply are Kansas City).

SOME ELF II ENHANCEMENTS

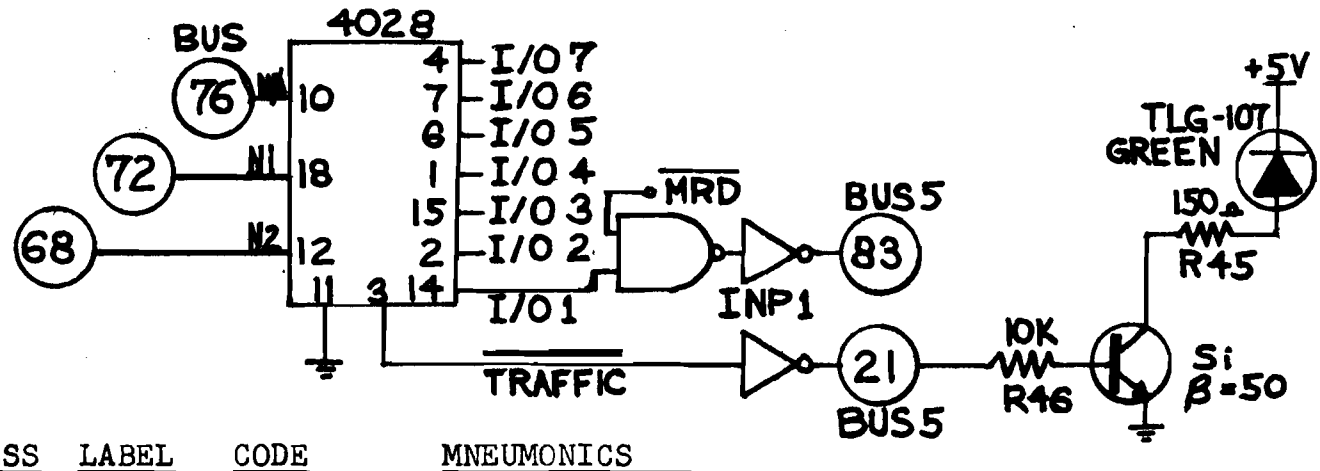
Tom Jones
1751 King Ave
Florence, S.C. 29501

I am expanding a new Netronics ELF II system on one of their excellent prototype boards, and am using the N-line decoder network shown below. The ELF II kit went together without a hitch, incidently. The quality of the parts and boards is quite good, I thought, although I would have liked a ceramic 1802CD. I also find the enclosed documentation adequate but very sparse. Considering the price, it seems fair value, especially as a springboard for experimenters.

The "TRAFFIC" indicator is in imitation of larger machines, and is a free spin-off of the unused "zero" output on the 4028 chip. It lights whenever there is activity on any of the seven I/O ports, and can be a reassuring sign things are going well. I mounted my green traffic led below the Q led, and transistor Q1 above it with R45, R46.

I hasten to report the pitfalls of modifying an existing design such as ELF II without careful research. After installing the N-line decoder, strange things began to happen. Data changed, programs ran away or crashed. I never even got a memory diagnostic to run long enough to check the supposed bad memory. The scope revealed that interrupts were bombing in on the 1802 suddenly and crashing the programs. Why? Because the 1861 video chip is jumpered to turn on whenever NO is generated, and my paper tape reader is on INP 3. The cure was to remove J9, install J10, and run pin 83 of bus 5 to the INP 1 section of the decoder circuits.

If some other ELF II user would like to copy the circuit diagram of the original Giant Board, it would help me to keep my system monitor board compatible with everyone else and out of trouble. I would be most grateful for any help.



ADDRESS	LABEL	CODE	MNEUMONICS
00		F810	LDI(STK)
02		AF	PLO F
03		EF	SEX F
04	START	6C	INP 4
05		64	OUT 4
06		2F	DEC X
07		3F04	BN4 (START)
09		6C	INP 4
0A		64	OUT 4
0B		370B	B4 *-1
0D		3003	BR(START)
0F		C4	
10	STK	00	DATA "00"

I use the above code on an ELF II as a subroutine within other programs. This routine allows the display to show hex keyboard data as it is being entered with the appearance of shifting to the left. This is more convenient for the operator, allowing correction of an error by simply keying in the correction before pressing the "input" key. It can save keystrokes too if you are quick.

The routine shown can be useful to load monitors to high memory.

SOME SIMPLE 1802 PROGRAMS

Jim R. Smith
4629 North Shore Dr.
Wichita Falls, Tx. 76310

Here is a trivial program.

```
XX00  F8  00  B5
  03  F8  00  A5
  06  E5
  07  64
  08  F8  7F  B6
  0B  26  96
  0D  3A  0B
  0F  30  07
```

When I first got my 1802 going, I ran the above program a lot. It outputs to the hex readouts the memory contents beginning on the page # at XX01. (The PGM will run on any page as is). I sometimes let it step through memory while I checked a program I'd written on another page but mostly I just let it run to make sure

it stayed busy. The display time is regulated by the byte at XX09. FF is twice as long and 3F is half as long.

Here is another simple program:

```
XX00  F8  YY  B6
  03  F8  ZZ  B7
  06  F8  00  A6  A7  A8
  0B  E6
  0C  72
  0D  57  17
  0F  28  88
  11  3A  0C
  13  00
```

It is a whole page transfer. "From" page is located at XX01 (yy) and the "to" page should be at XX04 (zz). When I want to relocate a page, I save it on cassette and play it back where I want it.

I never can find the paper it is written on, but I can always find the recorder! It can be written on any page without changes. (Obviously, it shouldn't be written on the page you are transfering to!)

PROGRAMMING THE DM8577 (32X8) PROM

G.F. Feaver
105 Townsend Ave.
Burlington, Ont.
L7T 1Y8

In IPSO FACTO #3 we discussed a hex display using a prom and standard seven segment display. The prom was programmed to take inputs from the present led readouts, thus utilizing the buffers provided and reducing the parts count.

The programmer is discussed below:

General information extracted from NSC Memory Data Book 1976 page 5-17 for the DM 8577 gives the following:

1. Apply steady state voltage to the chip: PIN 16=5.0V, PIN 8= Ground(0).
2. Disable outputs by applying logic 1 to "ENABLE" input PIN 15.
3. Address the word to be programmed with specific input voltage (0 or 1).
4. Select one output at a time for programming. All other outputs must be open.
5. Apply programming pulse of 20V (20-22V) and programming current of 150ma (100-200MA) for 10 to 20 MSEC.
6. Verify that output has truly been programmed.
7. Repeat steps 4 through 6 for each output of this address that must be programmed.
8. Advance to next address and repeat steps 3 through 7.

Programming unit description of operation see FIG 1

STEP 1. Turn on power to unit
2. Set S4, S5, S6, S7 to desired input address per truth table
3. Set S2 to desired output pin with the address selected (in step 2)
4. Set S3 to PROGRAM

PROGRAMMING THE DM8577 (32x8) PROM (CONT'D)

5. Turn S1 to ON position (allows burn current)
6. After about ½ second turn S1 to OFF
7. Turn S3 to "VERIFY" to check that bit has been programmed
8. Repeat items 3 through 7 for all outputs to be programmed "LOW" in selected address (Step 2)
9. Advance to next address and repeat steps 2 through 8

NOTE: When programming last row in Truth Table PIN 14 must be connected to VCC instead of 0 (GND). This row provided the lamp test and is not required for the HEX readout. It is a refinement.

EXPLANATION OF CIRCUIT

The PROM is programmed by burning out fusible links.

Power at 22.5 to 24 volts regulated is applied to voltage regulator X1 providing 5 volts (VCC) and 0 (GND).

X2-(7400)-provides an R-S flip flop and bounceless switch to turn on 3X-MONOSTABLE-(74121)-which provides the burn-in timing. It was found that 45 MSEC was required (instead of the specified 10 to 20 MSEC) probably due to rise time of the current power supply when voltage was suddenly increased.

When X3 goes, high it turns off the output transistor in X4 (75451) which previously had been shorting the 15V zener diode Z1 to ground thus limiting the output voltage of X5 (7805 voltage regulator) to 5 volts. When the shorting transistor in X5 turns off, the zener voltage (15V) is added to the nominal 5V to provide the 20 volt voltage used for driving the burn current through the fusible line in the PROM.

A led indicator is connected to the timed output of the mono to indicate that something is happening.

The verify led indicates that the programming burn has been successfully completed.

Rotary switch S1 is used to turn on the timer and complete the circuit for the burn current.

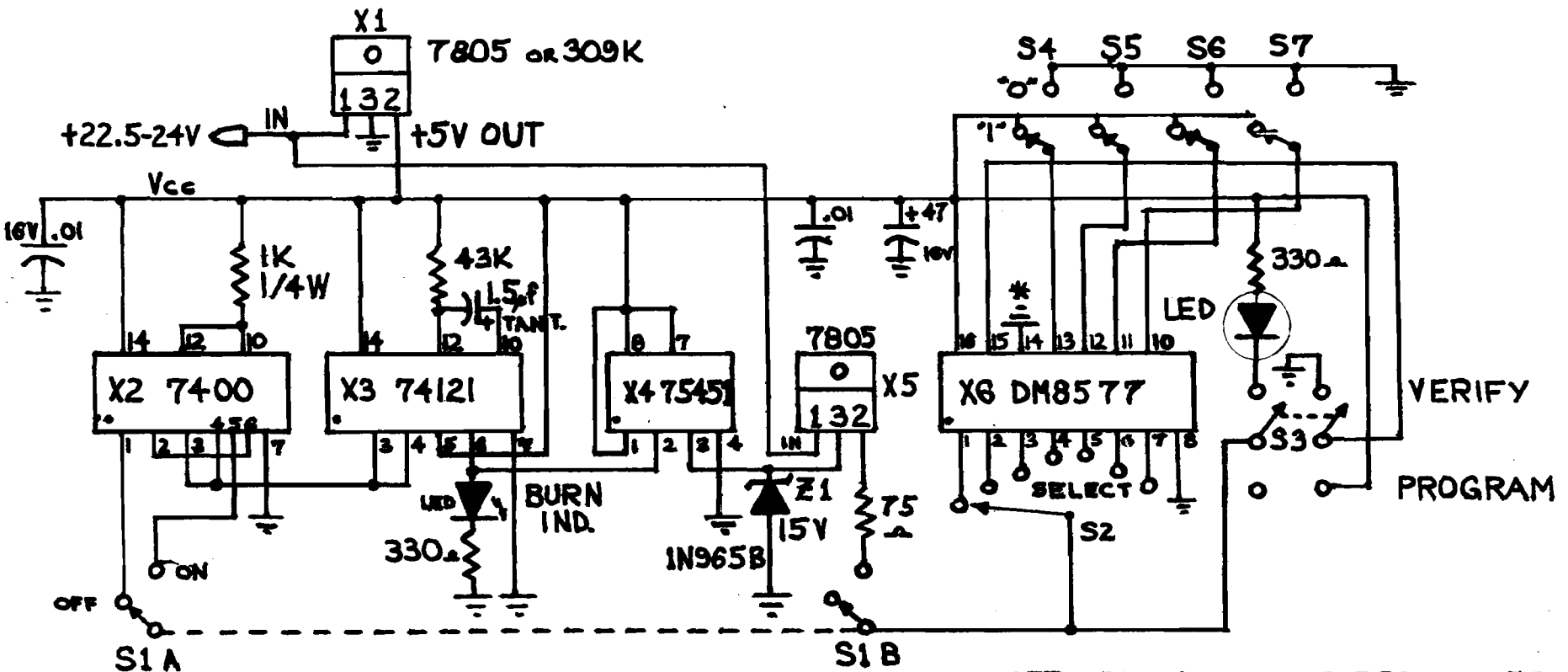
Rotary switch S2 is used to select the bit (PIN) to be programmed.

DPDT toggle switch S3 is used for program/verify.

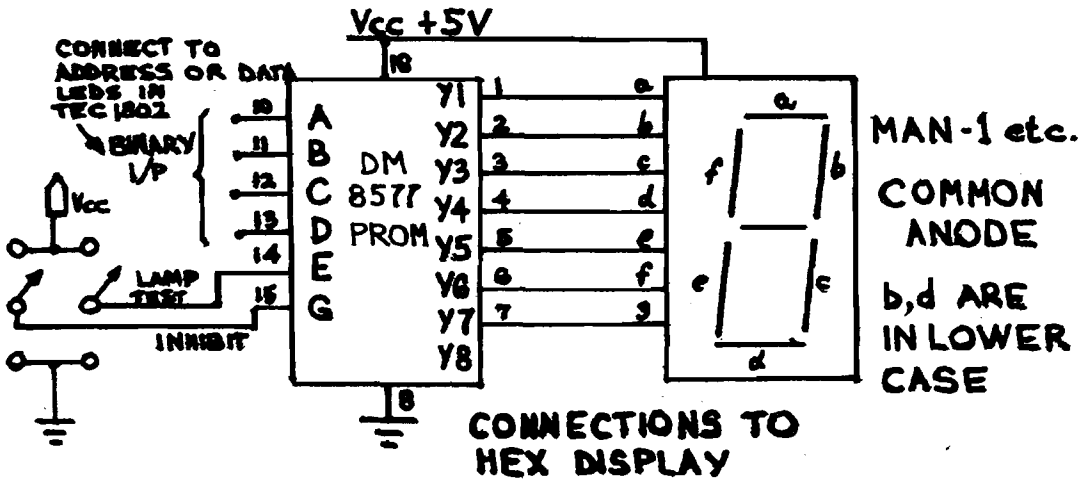
DPDT toggle switch S4 through S7 are used to select 1 or 0 for the input address.

It was not found necessary to use pull-up resistors to PINS 1 and 5 of X2 since no bounce was encountered.

CIRCUIT FOR PROM PROGRAMMER



* NOTE: PIN 14 MUST BE RECONNECTED TO Vcc FOR PROGRAMMING LAST ROW OF TRUTH TABLE



TRUTH TABLE AND PROGRAM FOR THE HEXADECIMAL DISPLAY

	INHIBIT	LAMPTEST	ADDRESS				DISPLAY	OUTPUTS							
			D	C	B	A		Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
PINS	15	14	13	12	11	10		1	2	3	4	5	6	7	9
	0	0	1	1	1	1	0	0	0	0	0	0	0	1	X
	0	0	1	1	1	0	1	1	0	0	1	1	1	1	X
	0	0	1	1	0	1	2	0	0	1	0	0	1	0	X
	0	0	1	1	0	0	3	0	0	0	0	1	1	0	X
	0	0	1	0	1	1	4	1	0	0	1	1	0	0	X
	0	0	1	0	1	0	5	0	1	0	0	1	0	0	X
	0	0	1	0	0	1	6	0	1	0	0	0	0	0	X
	0	0	1	0	0	0	7	0	0	0	1	1	1	1	X
	0	0	0	1	1	1	8	0	0	0	0	0	0	0	X
	0	0	0	1	1	0	9	0	0	0	0	1	0	0	X
	0	0	0	1	0	1	A	0	0	0	1	0	0	0	X
	0	0	0	1	0	0	Bb	1	1	0	0	0	0	0	X
	0	0	0	0	1	1	C	0	1	1	0	0	0	1	X
	0	0	0	0	1	0	Dd	1	0	0	0	0	1	0	X
	0	0	0	0	0	1	E	0	1	1	0	0	0	0	X
	0	0	0	0	0	0	F	0	1	1	1	0	0	0	X
	0	1	1	1	1	1	8	0	0	0	0	0	0	0	X

NOTE: B, D will be in LOWER CASE as b, d
NOTE 2: The Truth Table shown in IPSO FACTO #3 showed all inputs inverted

PING-PONG

Richard Delombard
802 Seneca Ave
Huron, Ohio, 44839

When the program bugs resident in your PL-1 cross disassembler finally overwhelm you, sit down and load this ping-pong program into your 1802 system. Challenge the toughest, meanest looking program bug to a duel in ping-pong. Since most program bugs are simple minded creatures, you should have no trouble handling it in the first game. If you don't have any program bugs ready to take the challenge, invite a friend to play. If you haven't any friends, try a left hand versus right hand tournament, and if that is too difficult, try a practice game against the flawless 1802.

The ball speed is adjustable by the data switches for matching the abilities of your opponents. The object is to swing your "paddle" when the "ball" is at your end of the court to hit it back to your opponent. If you swing too early or too late, the ball sails past you and the score is incremented and displayed. At the start of the game and after every point scored, a serve is initiated by either player swinging their paddle.

The program assumes the I/O hardware shown is Figs 1-3. The 8 data LED's are used for the ball bouncing back and forth across the court and also to display the score. The left 4 LED's is the

PING-PONG (CONT'D)

score in hex for the left player and likewise for the right 4 LED's. The two momentary push button switches serve as paddles to hit the ball back and forth. The 8 data switches are used to set the ball speed.

With proper modifications, other I/O hardware may be used with the program. Two keyboard switches can be used for paddles, hex digits to display the score and a TV display (with a CD1861) for the ball, paddles and a court.

Notice that a delay subroutine is called in several places in the program. This is accessed by changing the program counter to R5, which has been initialized to $006C_H$, the starting address of the subroutine.

A practice game against the 1802 can be incorporated by changing the 3E instruction at 0034_H to a 36 instruction. The 1802 will then play the right paddle and you will play the left. To switch sides with the 1802, leave the 3E in at 0034_H and change the 3C at 0051_H to a 34 instruction.

REGISTER USAGE

R8	$00E1_H$	Ball Position
	$00E2_H$	Delay
R9	$00E0_H$	Score
R3		Program Counter
R5		Subroutine Program Counter

I/O HARDWARE REQUIREMENTS

Input Port, N=4	Data Switches	Fig. 1
Output Port, N=1	8 LED's on DA0-7	Fig. 2
EF1, EF3	N.O. P.B. switches	Fig. 3

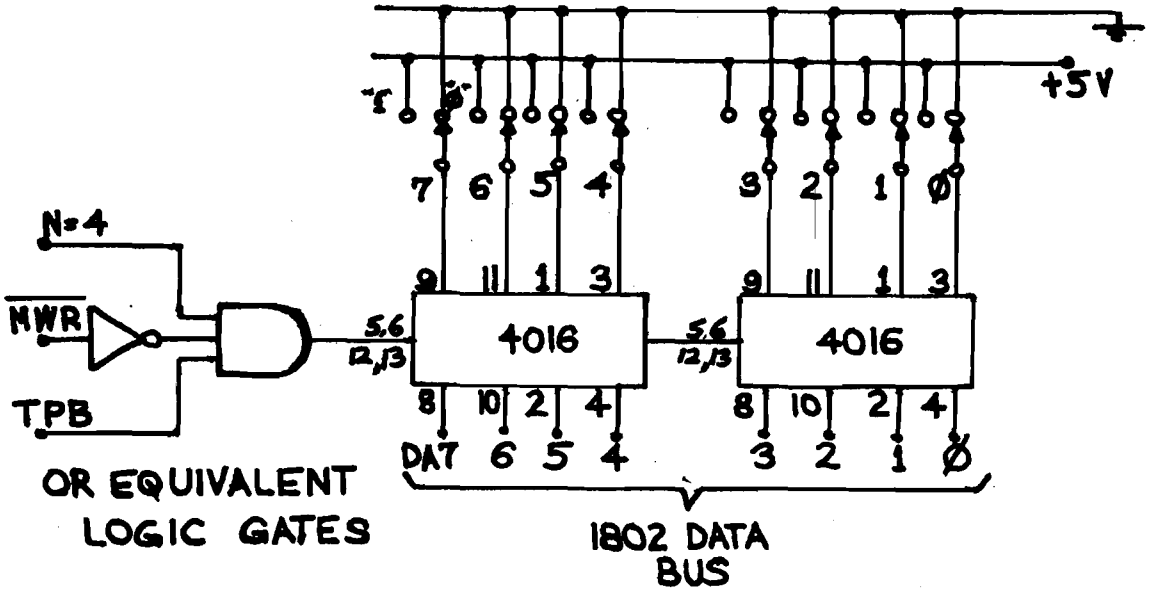


FIG. 1: DATA SWITCHES

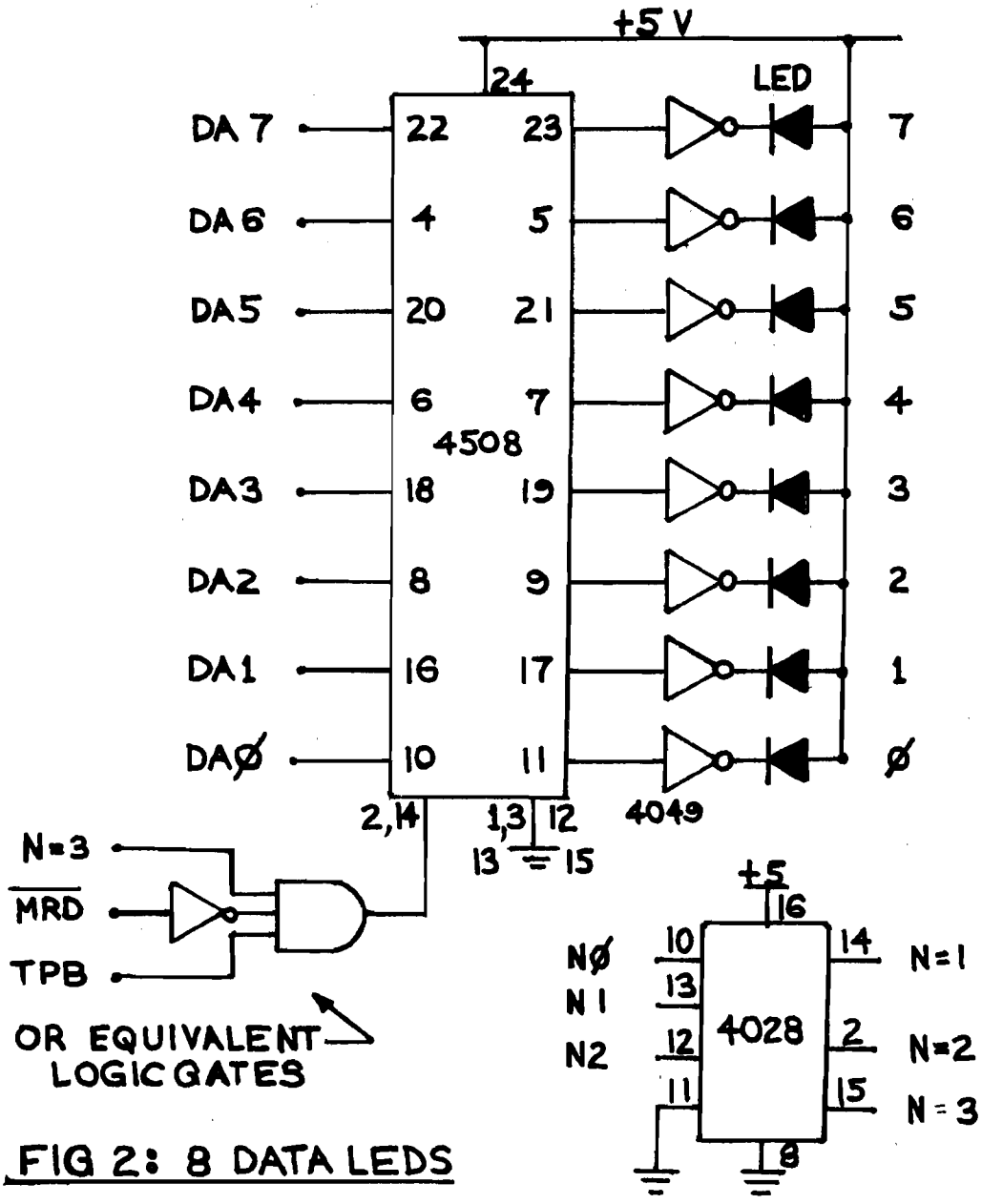


FIG 2: 8 DATA LEDS

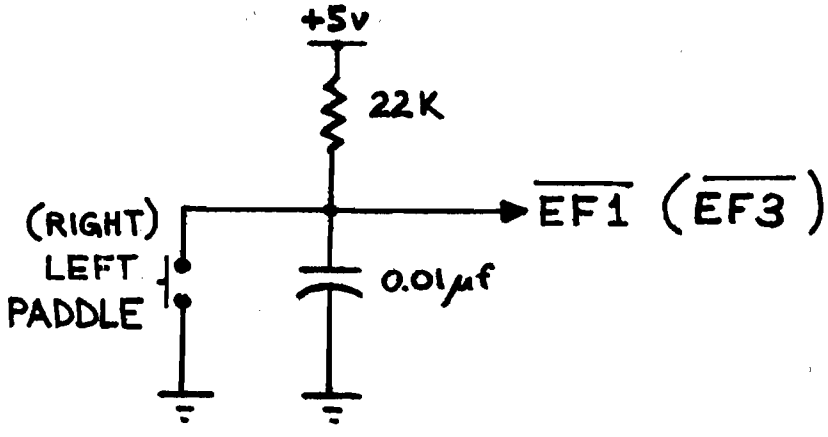


FIG 3: LEFT + RIGHT PADDLES

```
                     1 ..........................................................
                     2 ..       PING PONG GAME                                  ..
                     3 ..       WRITTEN BY RICHARD DELOMBARD                    ..
                     4 ..........................................................
0000 F8E0            5          LDI   A.0(SCORE)    ..SET UP R9
0002 A9              6          PLO   R9            ..AS SCORE ADDRESS
0003 F800            7          LDI   #00           ..CLEAR OUT
0005 B3              8          PHI   R3            ..HIGH ORDER
0006 B5              9          PHI   R5            ..BYTE OF
0007 B8             10          PHI   R8            ..WORKING
0008 B9             11          PHI   R9            ..REGISTERS.
0009 59             12          STR   R9            ..ZERO SCORE.
000A F80E           13          LDI   A.0(START)    ..SET UP P.C.
000C A3             14          PLO   R3            ..IN R3.
000D D3             15          SEP   R3            ..USE R3 AS P.C.
000E F8E1           16 START:   LDI   A.0(POSN)     ..SET UP R8 AS BALL
0010 A8             17          PLO   R8            ..POSITION ADDRESS.
0011 F86C           18          LDI   A.0(SUBEN1)   ..SET UP R5 FOR
0013 A5             19          PLO   R5            ..SUBROUTINE ADDR.
0014 C4             20          NOP                 ..SET UP
0015 C4             21          NOP                 ..ROOM
0016 C4             22          NOP                 ..FOR
0017 C4             23          NOP                 ..SPARES.
0018 E8             24          SEX   R8            ..POINT TO BALL POS REG.
0019 363A           25 SWT:     B3    RT            ..RIGHT SIDE SERVE?
001B 3C19           26          BN1   SWT           ..LEFT SIDE SERVE?
001D F880           27 LT:      LDI   #80           ..SERVE FROM
001F 58             28          STR   R8            ..LEFT SIDE.
0020 D5             29 DISP1:   SEP   R5            ..DISPLAY & DELAY SUBR.
0021 08             30          LDN   R8            ..GET BALL POSITION.
0022 F6             31          SHR                 ..SHIFT BALL TO RIGHT ONE.
0023 58             32          STR   R8            ..PUT NEW POSITION BACK.
0024 FA03           33          ANI   #03           ..CHECK IF NEW BALL POSITION
                    34 ..                           ..IS IN RIGHTMOST 2 POSITIONS.
0026 3220           35          BZ    DISP1         ..IF NOT,CALL DISPLAY AND DELAY.
0028 FA02           36          ANI   #02           ..IF SO, CHECK WHICH ONE.
002A 3231           37          BZ    BAK1          ..IF LAST POSITION,GOTO BAK1 AND
                    38 ..                           ..CHECK FOR PADDLE.
002C 3E20           39          BN3   DISP1         ..IF 2ND FROM LAST, CHECK FOR EARLY
                    40 ..                           ..PADDLE SWING.
002E 7B             41          SEQ                 ..IF EARLY PADDLE SWING, SET Q.
002F 3020           42 N3:      BR    DISP1         ..GOTO DISP1.
0031 D5             43 BAK1:    SEP   R5            ..CALL DISPLAY AND DELAY SUBR.
0032 3157           44          BQ    SCR1          ..IF Q  SET, GOTO SCORE (SCR1).
0034 3E57           45          BN3   SCR1          ..IF BALL NOT HIT, GOTO SCORE.
0036 08             46          LDN   R8            ..IF BALL HIT, GET BALL POSITION
0037 FE             47          SHL                 ..AND MOVE IT TO THE LEFT.
0038 303D           48          BR    DISP2         ..GOTO DISP2 TO MOVE BALL LEFT.
                    49 ..
003A F801           50 RT:      LDI   #01           ..GET BALL AT RIGHT SIDE IF
003C 58             51          STR   R8            ..RIGHT SERVE.
003D D5             52 DISP2:   SEP   R5            ..CALL DISPLAY & DELAY SUBR.
003E 08             53          LDN   R8            ..GET BALL POSITION.
003F FE             54          SHL                 ..MOVE BALL TO THE LEFT 1.
0040 58             55          STR   R8            ..PUT BACK NEW BALL POSITION.
0041 FAC0           56          ANI   #C0           ..CHECK FOR LEFT MOST 2 POS.
0043 323D           57          BZ    DISP2         ..IF NOT, CALL DISPLAY & DELAY.
0045 FA40           58          ANI   #40           ..IF SO, CHECK WHICH ONE.
```

```
LOCN OBJ CODE  STMT    SOURCE STATEMENT                1802 VER 1.6

0047 324E        59            BZ     BAK2          ..IF LAST ONE ON LEFT, GOTO CHECK
                 60 ..                              ..FOR PADDLE SWING.
0049 3C3D        61            BN1    DISP2         ..IF NEXT TO LAST, CHECK FOR
                 62 ..                              ..EARLY PADDLE SWING.
004B 7B          63            SEQ                  ..IF SO, SET Q.
004C 303D        64 N1:        BR     DISP2         ..IF NOT, GOTO DISP2.
004E D5          65 BAK2:      SEP    R5            ..CALL DISPLAY & DELAY SUBR.
004F 315B        66            BQ     SCR2          ..IF EARLY PADDLE SWING,GOTO SCORE.
0051 3C5B        67            BN1    SCR2          ..IF BALL IS NOT HIT,GOTO SCORE.
0053 08          68            LDN    R8            ..IF BALL IS HIT, GET BALL POSITION
0054 F6          69            SHR                  ..AND MOVE TO THE RIGHT
0055 3020        70            BR     DISP1         ..GOTO DISP1 TO MOVE BALL TO RIGHT.
0057 F810        71 SCR1:      LDI    #10           ..GET 1 FOR ADDING TO LEFT SCORE.
0059 305D        72            BR     SCRUP         ..GOTO SCORE UPDATE.
005B F801        73 SCR2:      LDI    #01           ..GET 1 FOR ADDING TO RIGHT SCORE.
005D E9          74 SCRUP:     SEX    R9            ..SET POINTER TO SCORE.
005E F4          75            ADD                  ..ADD CHANGE TO SCORE.
005F 73          76            STXD                 ..PUT IN NEW
0060 60          77            IRX                  ..SCORE.
0061 61          78            OUT1                 ..OUTPUT NEW
0062 29          79            DEC    R9            ..SCORE.
0063 3463        80 WT1:       B1     WT1           ..WAIT FOR PADDLES TO
0065 3665        81 WT2:       B3     WT2           ..BE RELEASED.
0067 E8          82            SEX    R8            ..SET POINTER BACK TO BALL POS.
0068 7A          83            REQ                  ..RESET Q.
0069 3019        84            BR     SWT           ..GO BACK AND WAIT FOR SERVE.
                 85 ..
006B D3          86 RET1:      SEP    R3            ..RETURN FROM SUBROUTINE.
006C 61          87 SUBEN1:    OUT1                 ..OUTPUT NEW DISPLAY.
006D 6C          88            INP4                 ..INPUT DATA SWITCHES FOR
006E 28          89            DEC    R8            ..DELAY.
006F B4          90            PHI    R4            ..PUT DELAY IN R4.
0070 24          91 LOOP:      DEC    R4            ..DECREMENT R4.
0071 94          92            GHI    R4            ..GET HI BYTE OF R4.
0072 3A70        93            BNZ    LOOP          ..IF NON ZERO, GOTO LOOP.
0074 306B        94            BR     RET1          ..IF ZERO, RETURN
                 95 ..
00E0             96            ORG    #E0           ..MOVE UP A WAY.
00E0 00          97 SCORE:     DC     #00           ..SCORE AREA.
00E1 00          98 POSN:      DC     #00           ..BALL POSITION.
                 99 ..
00E2            100            END

   0 DIAGNOSTICS GENERATED
  20 SYMBOLS

SYMBOL TABLE:
START    000E    SWT      0019    LT       001D    DISP1    0020
N3       002F    BAK1     0031    RT       003A    DISP2    003D
N1       004C    BAK2     004E    SCR1     0057    SCR2     005B
SCRUP    005D    WT1      0063    WT2      0065    RET1     006B
SUBEN1   006C    LOOP     0070    SCORE    00E0    POSN     00E1
```

ACE TUTORIAL

Rod Dore, Training Co-ordinator
660 Oxford Road, Unit 32
Burlington, Ontario L7N 3M1

The tutorial session has finally decided to undertake the design and construction of a KANSAS CITY STANDARD cassette interface. The basics of data recording up to the hardware and software to implement the storage and recovery of data will be studied. This project should be very interesting and useful and will also open the door to many other projects.

FIX FOR 1861 VIDEO DISPLAY JITTER

G. Fournier, VE2AIB
10155 Chambord
Montreal, PQ H2C 2R3

If your display sways from side to side and you are using the 3.58MH crystal, the clock frequency may be too far off from the 1.76MH frequency recommended.

I obtained a 8.810MH crystal (from ETCO) and divided by 5 to obtain a 1.762MH clock. Now the display is stable and a pleasure to watch.

(Ed. note: RCA recommends a 1.76064 MHz clock giving a line frequency of 60; a 3.579545/2 clock gives a line frequency of 60.99.)

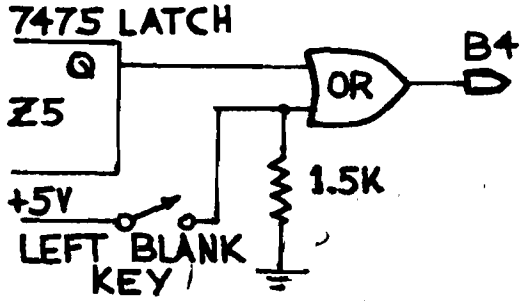
RADIO SHACK KEYBOARD ENHANCEMENT

G. Fournier, VE2AIB
10155 Chambord
Montreal, PQ H2C 2R3

To improve this keyboard and obtain HEX 10 11 ..... 1F output, make the following changes:

- Isolate the left blank key from the encoder board
- Cut the trace of the Q output of 1CZ5 and insert an OR GATE as per sketch.

To get HEX 10 11 ..... 1F DEPRESS LEFT BLANK & SHIFT KEYS and TYPE the letters that give HEX 00 01 .... 0F as per the output coding table.



NEW BASIC FOR COSMAC 1802 AVAILABLE

IPSO FACTO EDITOR:

I would like to invite our readers to participate in the testing and evaluation of an 1802 adaptation of Li-Chen Wang's "Palo Alto Tiny BASIC". If those interested give me 1) The beginning address of the area for the BASIC code and 2) the beginning address and length of the BASIC scratch area, I will furnish an assembly listing of the interpreter for a 50 cent (US or US stamps) mailing and reproduction charge. Details of the implementation are noted below.

Robert Edwards

T B I 1802

TINY BASIC INTERPRETER FOR THE COSMAC 1802

WRITTEN BY ROBERT EDWARDS

OAK RIDGE COMPUTER ASSOCIATION

P.O. BOX 295 OAK RIDGE, TN 37830

This basic interpreter is adapted from Li-Chen Wang's "PALO ALTO TINY BASIC" for the 8080 (PCC'S Reference Book of personal and home computing -- People's Computer Co.. Menlo Park, CA). All the features of that implementation are retained in the 1802 version:

DIRECT
COMMANDS: LOAD, DUMP, NEW, RUN, LIST

INDIRECT
OR DIRECT: LET, IF, FOR/NEXT, PRINT, REM, STOP, POKE
COMMANDS

INDIRECT
COMMANDS: GOSUB/RETURN, GOTO, INPUT

FUNCTIONS: ABS, RND, PEEK

ARITHMETIC: 16-BIT INTEGER

VARIABLES: 26 16-BIT SCALARS (NAMED A THROUGH Z) AND AN ARRAY (NAMED @) OF 16-BIT SCALARS IN THAT PORTION OF THE TEXT AREA NOT OCCUPIED BY BASIC STATEMENTS

MEMORY USAGE

2000-2FFF	TINY BASIC SCRATCH AREA (VARIABLES, INPUT LINE, AND STACK IN THE FIRST PAGE; THE ARRAY STARTING AT THE SECOND PAGE AND WORKING UP; AND THE BASIC STATEMENTS AT THE END OF THE SCRATCH AREA) (SET UP FOR 4K)
F000-F7FF	TINY BASIC CODE (2K BYTES)

NETRONICS MONITOR DISASSEMBLED

Kenneth Mantei
1254 E. 26th Street
San Bernardino, California 92404

Those of you Elf II owners who have the "Giant" board may have occasionally wondered how the monitor works. The cassette save and load routines are interesting because they are not trivial. Understanding how the initialization proceeds will enable you to predict the page of your RAM that will have its FE and FF bytes mysteriously modified (by the monitor). This turns out to be page 03 in my five page RAM! Some of the comments below are more detailed than others. Presumably those who are advanced enough to benefit from the disassembly will be able to see what is unsaid.

LABEL	ADDRESS	BYTES	COMMENTS
INIT	YZ 00	90 A1 B3 B4 B5 B6 B7	YZ → hi order R3-7 & lo order R1.
(PC=R0)	07	F8 FF A2 E2 21 81	Establish scratch area by attempting
	0D	B2 80 52 F3 3A 08	store & retrieve at FFFF, FEFF, FDFF etc. until successful. Scratch address → R2.
	13	F8 38 A3	R3=YZ38 MAIN address.
	16	F8 29 A4	R4=YZ29 INPUT address.
	19	F8 33 A5	R5=YZ33 DISPLAY address.
	1C	D3	Jump to MAIN.
DISP & INP	1D	3F 1D	Wait for Press.
(PC=R0)	1F	22 8A 52 64	RA.0 displayed & saved at scratch FE.
	23	37 23 6C	Whatever keyed when Press released
	26	30 1C	is displayed & saved at scratch FF. Return to MAIN
	28	D3	Exit INPUT & return.
INPUT	29	3F 29 37 28	Wait for Press & unpress.
(PC=R4)	2D	6C 64 22	Input, display & save byte at
	30	30 28	scratch area. Return to MAIN.
DISPLAY	32	D3	Exit DISPLAY & return.
	33	22 52 64 30 32	Display & store D reg in scratch FE.
(PC=R5)			
MAIN	38	D4	Input f 'n 00-05 & store in scratch FF.
(PC=R3)	39	F8 4D F4 A6	4D + f 'n → R6.0=address of f 'n subroutine.
	3D	02 FD 05 33 47	If legit f 'n jump to STARTADD.
	42	F8 EE D5 7B 00	Display EE, turn on Q, & quit.
STARTADD	47	D4 8A D4 AA	Starting Address → RA
	4B	06 A3	Get f 'n sub address from R6 & Jump
	4D	53	to: RUN YZ 53
	4E	58	EXAMINE 58
	4F	5D	CHANGE 5D
	50	78	SAVE 78
	51	C8	LOAD C8
	52	F6	SEARCH F6
RUN	53	9A 80 8A A0 E0 (D0)	Start Add → R0; X=0, Run.
EXAMINE	58	D0	Jump to DISP & INP to show & inc. lo order address from RA while key pressed. When
	59	4A D5 30 58	released, program byte displayed. Repeat.
CHANGE	5D	7B	Turn on Q light.
	5E	D0	Jump to DISP & INP to show & inc. lo order address from RA while key pressed. When released, last byte keyed in is input.
	5F	D5 5A 1A 30 5D	Byte displayed & placed in program replacing old byte. Repeat.
	64	D3	Exit CYCLE & return.
CYCLE	65	7B	Turn on Q. DF=0 for "0" bit, 1 for "1".
(PC=R6)	66	F8 1D	Half cycle length (decimal 29) → D
	68	3B 6D	Jump to LEN if DF=0 (minus).
	6A	F8 07 1D	H.C.L. (decimal 7) → D; Inc RD on "1" bits.
LEN	6D	52	Store H.C.L. in scratch FF.
	6E	FF 01 33 6E	Decrement by one until reach zero.
	72	39 64	Return if Q off.
	74	7A 02	Turn Q off. Load H.C.L.
	76	30 6E	Count down once again, w/ Q off.

```
SAVE      78    F8 8D A1       Set up for later branch just
                               before HEADER.
ENDADD    7B    D4 73          Hi order ENDADD→scratch FF,
          7D    D4             lo order   "  "   "   FE,
                               pointer left at FE.
          7E    8A F5 AC       # Bytes, less than full block,
                               to record ⟶ RC.0
          81    12             Pointer to scratch FF.
          82    9A 75 FC 01 BC # Full + partial pages to save or
                               load →RC.
          87    D0             Key press displays lo order start
                               add & begins recording or load.
          88    F8 65 A6       Set R6 to point to CYCLE.
          8B    81 A3          Jump to HEADER in SAVE mode, to
                               Go in LOAD mode.
HEADER    8D    F8 80 BD       80 → RD.1, 14 sec counter.
          90    FF 00          DF →1.  Set up for "1" bit.
          92    D6             Output via Q line one "1" bit,
                               adding 1 to RD.
          93    9D 3A 90       Repeat until 14 sec of "1" bits
                               completed=cassette save header.
BYTE      96    8A D5 7B       Get lo order address & display.Q on.
          99    4A             Load byte to record & inc pointer.
          9A    BB             Save byte in RB.1
          9B    FC 00          DF→0 to set up "0" start bit.
          9D    F8 09 AB AD    "09"→RB.0, RD.0
BIT       A1    D6             Output via Q line, one bit sent.
          A2    2B 8B 32 AB    Jump to PARITY if start bit
                               plus 8 bit byte completed.
          A6    9B FE BB       Shift byte left; bit to be sent
          A9    30 A1          →DF.  MSB sent first, LSB last.
PARITY    AB    8D F6 D6       Send parity bit, "0" for even
                               "1" for odd.
          AE    2C             Decrement bytes-to-be-sent counter.
          AF    9C 3A 96       Jump to BYTE unless all sent.
          B2    D6 D6 D6 D6 30 38  Send 4 garbage end bits.→MAIN.
          B8    1D             Increment reg RD.0 if "1" bit.
          B9    D3             Return to LOAD.
TEST      BA    F8 0D          BC    35 BC    Tests cassette signal for "0" or
                               BE    35 B8    "1" bit. If "0", DF=0 & RD is
                               C0    FF 01    not incremented. If "1", DF=1 &
                               C2    33 BE    RD is incremented. Returns to
                               C4    3D C4    MAIN.
                               C6    30 B9
LOAD                           C8    F8 CD A1 Set up for branch just before HEADER.
                               CB    30 7B    Jump to ENDADD to calc # bytes
                                              to be loaded,→RC.
GO                             CD    F8 BA A7 Set R7 to point to TEST.
                               D0    F8 F9 BD F9→RD.1 (7 less than zero).
                               D3    D7       Test cassette header for "1" bit.
                               D4    3B D0    Start TEST over until "1" header
                                              bit is seen.
                               D6    9D 3A D3 Keep trying until 7 successive
                                              header bits read correctly.
READ                           D9    D7 33 D9 Wait until see "0" start bit.
                               DC    F8 01 BD AD RD.0 & RD.1 initialized to 01.
                               E0    D7       Test bit (inc RD.0 if "1").
                               E1    9D 7E BD Ring shift left RD.1, storing
                               E4    3B E0    bits in RD.1 until byte complete.
                               E6    D7       Test parity bit (will be "1" if
                               E7    8D F6    even # of 1's in byte). Quit if
                               E9    33 45    parity bit + # 1's in byte +
                                              original 01 accumulated in RD.0
                                              is odd.
                               EB    9D 5A    Write byte into memory.
                               ED    8A D5    Display lo order address of byte
                                              just written.
                               EF    1A       Increment pointer to next address.
                               F0    2C       Decrement bytes-to-be-read counter.
                               F1    9C 3A D9 Jump to READ unless all bytes
                                              loaded.
                               F4    30 38    Return to MAIN.
SEARCH                         F6    D4       Input byte to be sought.
                               F7    4A       Load byte to check.
                               F8    F3 3A F7 Repeat until find byte.
                               FB    2A       Dec. address back to where
                                              byte was found.
                               FC    9A D5    Get & display hi order address.
                               FE    30 58    Go to EXAMINE.
```

Partial Display Subroutines for 256 Byte Elf Systems

Ken Mantei
San Bernardino, Calif.

The graphics test program displaying the Enterprise Spaceship (Popular Electronics, p. 42, 44, July 1977, and in Netronics Assembly Manual) also shows the program bytes. If the graphic-modifying MAIN subroutine is not needed, space becomes available for modified INTERRUPT subroutines which display only the graphic.

Running from 0050-00FF, the graphic consists of 22 8-byte rows. Each row is scanned 4 times for a total of 88 desired graphic scans. To blank the screen during the first 40 (=128-88) scans, two 8-byte rows of zeros must be scanned 20 (=14 in hex) times. These are conveniently supplied in the original programs at 0400-04FF; they could however appear anywhere on the same page as the graphic.

In the Top Blank program, the INTERRUPT subroutine puts half the number of blank scans in an unused register which is decremented and tested during blank scans. When this register reaches zero, the program jumps to load the graphic display address into R0. Two zero rows seem to be necessary since the decrementing, comparing, and repointing of R0 takes more time than is available in one between-scan period.

To display the graphic with unused lines at the bottom blanked, it is necessary to place the graphic in memory so that it can be immediately followed, on the same page, by an 8-byte row of zero bytes. The graphic, originally at 0050-00FF, must be moved to 0048-00F7. 00F8-00FF must be zeros to be repeatedly scanned during blanking.

Program Bottom Blank implements this. A minor complication arises, due to the fact that during the last four scans the video chip pulls the display status line low. In the original program, the 3C instruction sensed this only as the last 4 scans were completed. In Bottom Blank this "almost-done" signal is sensed with 3 scans still to go. So the FOUR subroutine is added to keep repointing R0 to zeros until the display status line again goes high, signaling that the display window in finally closed. Without FOUR, the bytes following the zero row get displayed.

Once the logic of these modifications becomes clear, one should be able to write interrupt routines to blank the screen at the top, bottom, or top and bottom. Of course, users with more than one page of memory can dedicate one page for display, blank anywhere easily with zeros, and use the standard interrupt routine.

INTERESTING COMPUTER-ORIENTED ARTICLES

(extracted from Radio-Electronics,1978)

C. John Williams
323 Walkers Line
Burlington, Ont.
L7N 2C8

Month	Pg.	
Jan.	68	Z-80 u-P Instruction Set
Feb.	45	Hobby Computers Cos' hardware, names & addresses.
"	72	Z-80 Addressing modes.
Mar.	68	Z-80 Interrupts.
"	70	Hobby Computer Cos.,(conclusion).
"	100	Teaching digital counter IC's to count.
Apr.	48	All about the S-100 Bus.
"	80	Z-80 interfacing.
May	44	Introduction to peripherals & their interfacing.
"	82	6502 u-P & it's instruction set.
Jun.	37	Your Own Computer - 32p. on who, what, where, how, etc.
Aug.	38	Video Modulator - roundup of available units, Mfrs.
"	64	Intel 8253 programmable timer IC.
"	72	The REF-02 - a temp.sensor & precision V.Regulater IC.
Sep.	74	Intel 8253 timer IC - it's characteristics.
Oct.	78	6800 u-P hardware & software aspects.
# Dec.	45	MM57109 - an NCM,(Number-Oriented u-P), a "number-cruncher" for a CDP1802 u-P.

Top Blank Program

Label	Address	Bytes	Comments
	00 00	90 B1 B2 B3 B4	B4 unnecessary in this mod.
	05	F8 39* A3	*Address changes from P.E.
	08	F8 3F* A2	original Graphics Test Program
	0B	F8 11 A1 D3	
RETURN	0F	72 70	
INTERRUPT	11	22 78 22 52	
	15	C4 C4 C4	
			Bytes from 0018 to 003F
			are different from original
			Graphics Test Program
	18	F8 14 A5	#Blank scans/2 (in hex)→R5.0
	1B	F8 00 B0	Address of 2 zero lines →R0
BLANK	1E	F8 40 A0	
	--	DMA zero row 1	
	21	25 85 32 27	Countdown blank scans by 2 and
	--	DMA zero row 2	jump on zero to PIC to display
	25	30 1E	graphics beginning at 0050
	--	DMA zero row 2	
PIC	27	F8 50 A0	
REFRESH	2A	E2	
	--	DMA Scan 1	Scan each 8-byte line
	2B	E2 20 A0	four times as in original
	--	DMA Scan 2	Graphics Test Program
	2E	E2 20 A0	
	--	DMA Scan 3	
	31	E2 20 A0	
	--	DMA Scan 4	
	34	80 3C 2A	Check for end of display
	37	30 0F	window.
MAIN	39	E2 69 30 3B	Do-nothing MAIN loop
	30	XX XX XX	3D to 3F is stack area
ZERO	40	00 00 00 etc.	Bytes 0040 - 00FF containing
GRAPHIC	50	7B DE 0B etc.	graphic same as original P.E.
			Spaceship Program

Bottom Blank Program

Label	Address	Bytes	Comments
	00 00	90 B1 B2 B3 B4	B4 unnecessary in this mod.
	05	F8 39* A3	*Address changes from P.E.
	08	F8 3F* A2	original Graphics Test Program
	0B	F8 11 A1 D3	
RETURN	0F	72 70	
INTERRUPT	11	22 78 22 52	
	15	C4 C4 C4	
	18	F8 00 B0	
	1B	F8 48 A0 E2	Spaceship graphic moved to
REFRESH	1F	80	begin at 0048 instead of 0050
	--	DMA scan 1	
	20	E2 20 A0	
	--	DMA scan 2	Scan each 8-byte line
	23	E2 20 A0	four times as in original
	--	DMA scan 3	Graphics Test Program
	26	E2 20 A0	
	--	DMA scan 4	
	29	FB F0	Makes D=0 if last line.
	2B	3A 1F	Get new display line if D≠0.
	2D	80	Save in D address of 8 zero
	--	DMA zero row	bytes that must immediately
			follow last line of graphic.
BLANK	2E	20 A0	Repoint to zero row until
	30	3C 2E	EFI shows we're in last 4 scans.
	--	DMA zero row	
FOUR	32	20 A0 34 32	Repoint to zero row until
	--	DMA zero row	EFI shows last 4 scans done.
	36	30 0F 00	
MAIN	39	E2 69 30 3B	Do-nothing MAIN loop.
	3D	xx xx xx	3D to 3F is stack area. 40 to 47 is unused.
GRAPHIC	48	7B DE 0B etc.	Bytes 0048-00F7 contain graphic found as 0050-00FF in original.
ZERO	F8	00 00 00 etc.	Zero row for blanking.

ERRATA

The software for the CMOS 16X32 video system, IPSO FACTO #8, page 12 has two errors.

LOCATION	WAS	SHOULD BE
8386	9C	9E
8389	BC	BE

RUNNING THE "IPSO FACTO" STANDARD CASSETTE SOFTWARE ON THE ELF

Bernie Murphy
102 McCrany St
Oakville, Ont.
L6H 1H6

Some of you may be interested in running the cassette software as described in IPSO FACTO issue #8, page 37 but have hesitated as the timing constants in the article were for a 1 MHz clock.

There are two solutions to the above problem.

SOLUTION 1) Feed an external 4800 Hz clock to one of the EF lines and change the delay code as follows:

```
0065  DD      DELRTN:  SEP   R13     ..Actual Delay is 16 Clock
                                       Pulses, But One May Pass
                                       Undetected
0066  F80F    DELAY:   LDI   15      ..This Delays
0068  3668    DELOOP:  B3    *       ..For 15 Clock Pulses
006A  3E6A             BN3   *
006C  FF01             SMI   #01     ..Delayed One Bit
006E  3A68             BNZ   DELOOP  ..Yet?
0070  3065             BR    DELRTN  ..Yes, Return
0038  F808             LDI   #08     ..Delay 8 Pulses
```

This new delay routine is 3 bytes larger, so you will have to relocate the cassette dump routine--horrors!

SOLUTION 2) Change the timing constants.
Eg. The ELF CPU clock is 1.79 MHz

for 1 MHz, 62 loop delays are required
for 1.76 MHz, 62 x 1.79
$\simeq 111$
$\simeq 6F_H$ delays

∴ code changes are:

0038	F837	½ BIT TIME
0066	F86F	1 BIT TIME

I have not actually tested the above calculations in practise. You may have to "fine tune" the above constants to get exactly 300 BPS.

If anyone gets this software going, I would be pleased to hear from them.

References:

1. IPSO FACTO Standard for KC Tapes, IPSO FACTO #8, page 37.
2. Cassette Test Routines, IPSO FACTO #3, page 30.

NETRONICS' "GIANT BOARD" NOTES

I acquired Netronics' "Giant Board" for my ELF II a few months ago. After assembly, I couldn't make it work. It took me several weeks of checking and rechecking before I realized I didn't have enough power-even though Netronics claimed no extra power was required for the board. I already had 1 amp on my basic Elf.

I just mention this in case someone is presently experiencing the same problem I had.
G.R. Gilbert, 304 Vassal, Drummondville, Que. J2B 5H3

I have a Netronics ELF-II with the Giant Board and have the following experiences to share. Initially everything worked well but after adding the Giant Board, the 1861 video controller didn't work. A phone call to Netronics yielded the suggestion to try the modification described as step #3 in version #1 for my version #2 board. After making the mod.., the video resumed working. All of the neighborhood 7 year olds are now having a good time with Dan Rubis' Mousetrap (IPSO FACTO #6). The other problem that I had was that the cassette I/O on the Giant Board wouldn't read back a program which it had written. I am using the least expensive Sears unit that has a tape counter. The difficulty turned out to be the limited high-frequency response of the record/playback combination. The cure was to put a 0.002 ufd capacitor in series with the ear-phone output of the recorder. The capacitor functions as a high-pass filter (reducing the low-frequency response to balance the high-frequency loss of the system). The performance of the recorder is now fine.
Joe Rothberg, 2908 New Castle Ave., Silver Spring, MD, 20910, USA.

LETTERS TO THE EDITOR

Dear Bernie,

Just a quick note. I just received my Oct. issue of IPSO FACTO and in reading over the software for CMOS 16-32 Video System, I realized I had sent you my original program which contains two errors.
1. location 8386 9C should be 9E
2. location 8389 BC should be BE

The program works okay until you attempt to scroll with cursor on line # nine. Everything will scroll up but the cursor will appear on the bottom line (#16) rather than line #8. My appologies for not sending my revised program. I'm presently working on a program "HEX PAWN" using artificial intelligence where the computer learns from its mistakes and progressively gets smarter. I am trying to get it into 1K of RAM. (Just a challenge as I have 5K of RAM).
Sincerely, George E. Millar, P.O. Box 1412, Parksville, B.C. VOR 2SO

Dear Bernie,

I just received Issue #8 and it was fantastic! One note on my program - Variations on a theme - which I'm sure was picked up. I show a loop from Memory location 13_H Object code 85 to ML 01_H. When actually it is from ML $14/15_H$ Object code 32 01. Also I've been able to run this program on any page of memory by jumping to it from an initialization routine which sets the high order bytes of the registers to the page address.

I really appreciate the publication of the "IPSO FACTO" Standard for Kansas City Tapes. I agree that it is very important that we be able to exchange tapes with other 1802 users. UNFORTUNATLY I'm afraid I won't be able to use the Standard because at least at this point in time, I have no idea how to determine the values needed to replace the values listed for a 1MHz clock. Running an Elf II, as many other members do, I'm locked in to a 1.78MHz clock unless I want to give up my video display with the 1861 chip. Would it be possible to publish the process (other than Trial and Error) to determine the correct value for a particular clock speed or simply a table of values for the "standard" clock speeds of the available 1802 based micro's? (Will do. B.M.)

I know of one 1802 system up and running with 16k of dynamic memory. The developer (who is a dynamic memory designer!) will be, hopefully, marketing his design in the near future. I'll let you know more when I can. He has also done some remarkable extensions to TINY BASIC AND MAY TRY TO WORK OUT SOMETHING WITH Tom Pitman to market them (sorry about the caps, I forgot to hit the key). Thanks again for a super issue, David A. Hersker, 439 S. River St., Apt #5, Wilkes-Barre, PA 18702, USA.

Dear Bernie,

The November 20, 1978, issue of EDN magazine has a section on microprocessors. The page about the 1802 also mentions the 1804 and 1806. The 1804 includes 2K bytes of masked ROM, 256 bytes of RAM, and a timer. There are 13 more instructions than in the 1802, some of which are for the timer. The additional instructions use the previously used opcode (68 hex) for the first byte. The second byte specifies which of the new operations is to be performed. A version of the 1804 with erasable PROM instead of masked ROM is under development.

The 1806 is listed as a "possible future enhancement" of the 1802. It would have instructions for conventional subroutine calls and 16-bit load immediate instructions for registers 0 thru F.

Your mention of floppy disks caught my interest. Does any disk operating system (DOS) exist for the 1802? I have designed a rather ambitious DOS for possible implementation on a personal computer system. So far I have mostly specified the file system and the interface between user programs and the operating system. No coding yet, since among other things, I have not decided which processor to write it for (nor do I presently have any disk drives).

I now have my "semi-tiny BASIC" for the 1802 running under a simulator program. (My ELF II needs a terminal, more memory, and paper tape or cassette before "semi-tiny" will run there. All of these are presently in the works.) "Semi-tiny BASIC" is similar to other tiny BASICs, but includes FOR-NEXT loops and several built-in functions, and uses 4K of memory. The interpreter uses no three-cycle instructions so that it can run with the graphics chip on the ELF II without a jittery display. I do not know if any other 1802 BASICs, tiny or otherwise, have this feature. Sincerely,
Jim Howell, 5472 Plays Del Ray, San Jose, CA, 95123.

Mr. York,

I recently purchased an 1802 based microcomputer from QUEST Electronics. It has 256 bytes of RAM and a 32 byte ROM monitor. It also has a video display interface chip (CDP1861), power supply and on board provisions for memory and I/O expansion. I would like to become a member of your ACE club to permit me to receive IPSO FACTO publications.

I hope I may contribute to your publication, although I am an Electronic Engineer, microprocessors and software programming is a new challenge to me. I am currently working at the Jet Propulsion Laboratory in Pasadena in the Spacecraft Data Systems Group. I am presently designing the command decoder subsystem for Gallileo Mission (Jupiter Orbiter Probe '78) on the command and data system. This is a distributed data system utilizing radiation hard 1802 processors. The spacecraft will be launched off the space shuttle. After the design phase, I will help write some test software for Jamous subsystems. I'm eager to obtain any 1802 related articles and so I hope to hear from you soon. Sincerely, Robert S. Valencia, 266 N. Wilson St., Sierra Madre, Calif., 91024.

Dear Sirs,

My system consists of a Quest Super Elf which has been altered and 32K of (seals) memory.

I am now in the process of interfacing a computer controlled cassette and will gladly provide a more detailed discription of my toy. (Please do! B.M.)

Gentlemen, if at all possible, don't send the newsletter by surface mail. It takes about 6 months.

Looking forward to hearing from you. Sincerely, C. W. Vlaun, P.O. Box 624, Serce Colorado, Aruba.

Dear Bernie,

RE: RADIO SHACK KEYBOARD , KEY

Further to my letter of November last, I wish to report the following:

The manager of the store, Mr. Normanoin took it upon himself to see a T.V. repairman and get the work done, pay for it out of his pocket and advise me to pick it up.

The serviceman heated the finger on the underside of the encoder board until withdrawn replaced on the proper side of the plastic spacer and resoldered Voila!

This incident shows the good will, guarantee and service after sale of RADIO SHACK!!

If I did not benefit by your intervention, I am sure other members of the club will should they be faced with similar situations.

Thanks for all Bernie & long life to the club. Yours,
G. Fournier, 10155 Chambord, Montreal, Que., M2C 2R3.

Dear Bernie,

Just a note to let you and the rest of ACE know how I'm doing with my system.

I now have 9K (8K+1K) running. A printer and an image sensor are planned before Jan '79.

The 8K board was S-100 but isn't anymore...I used it only as a place to put 64 2102's!

LETTERS TO THE EDITOR CONT'D

The image sensor is only theory yet but an article is definitely in the works. (Any word on the Netronic's Elf-II Newsletter?) Yours truly, Darcy Roberts, 660 Laurier, Brockville, Ont. K6V 5X8

George York,

Thanks for your letter telling us of your activities. We at Quest Electronics want to be of service to the 1802 community and invite your readers to send us questions about the types of products you would like to have; and any other questions you like. Enclosed is some literature about the products we are now offering to the COSMAC experimenter.

We also have a publication, QUESTDATA, which provides a forum for users and prints interesting software and other information.

Enclosed is a NEWS RELEASE with information we hope you will mention to your readers. Best Regards, Bill Haslacher, Editor, Questdata, P.O. Box 4430, Santa Clara, CA, 95054.

QUEST ANNOUNCEMENT:

A new club to support the RCA 1802 COSMAC is QUESTDATA. Owners of Elf, Super Elf, Elf II, COSMAC VIP, COSMAC Development System or Homebrew 1802 will find many programs, applications, and experiments for their microcomputer in each issue of QUESTDATA.

QUESTDATA will be showing the complete RCA instruction set and how to build interesting programs for: graphics, control, games, and business purposes. Coverage will be given to Tiny BASIC, Elf Expansion possibilities (memory, cassette I/O, etc.), light pens, reader questions and music programs.

QUEST Electronics, publishers of QUESTDATA and maker of the Super Elf (one-board computer with graphics), wants to give the best possible support to COSMAC systems. QUEST wants users to learn and grow in knowledge with their Elf. With a Super Elf and QUESTDATA subscription, you can start with a small investment in computers (the Super Elf kit is $106.95; QUESTDATA is $12 a year) and grow with your knowledge into the system configuration of your choosing. The choices for Elf expansion are many, from simpler controller applications--a long-distance telephone, chess timer program appeared in the first issue--to complex systems using 4K or more memory.

QUESTDATA offers users the growth possibilities which all Elf systems provide. The $12 monthly QUESTDATA--P.O. Box 4430, Santa Clara, CA 95054 will give your Elf's memory some microcomputer brain food. Foreign subscriptions, with the exception of Canada and Mexico, are $6 extra for mailing.

ITEMS FOR SALE

1. For Sale: New ASCII keyboard, never used, assembled and tested, with detailed manual, asking $35. Jim LaVeck, Route 1, Box 180, Dexter, New York, 13634 USA, 315-639-6383.

2. For Sale: Complete Elf II System, up and running. Includes Elf II, 4 amp power supply, Giant Board, Kludge Board, and cassette recorder, all in perfect working order. Also, over 400 pages of programs and applications, very extensive documentation, $175 US. Jim LaVeck, Route 1, Box 180, Dexter, New York, USA 13634, 315-639-6383

ITEMS FOR SALE CONT'D

3. (3)Transformers, Pri taps for 110, 115, 105, 125. Sec: 5V@3A, 6.3CT@4A, 700VCT@.8A. Solid high quality military surplus, epoxy sealed for quiet operation. $10 ea.

(1)Hex-encoded Keyboard using desk calculator keyboard and TTL decoder logic on wire-wrap panel. Outputs 4 bits hex plus 4 controls (RUN, WAIT, LOAD, RST) $6.00

(2)Wirewrap panels. "Dead-bug" type (I.C. inserted legs up and tack soldered on comp side) holds 200 14 pin I.C.s. No wire on these new panels, a very few easy to cut etches on boards. $5.00 ea.

(1)Floor optic assy. Provides ideal light source for 80 column card reader. (End feed machine, 12 light points precisely spaced, with lamp. $2.00

(4)8" Floppy Disks. $2.00 ea.

LETTERS OF CONTACT

My Netronics' "Giant Board" works fine except that I cannot read cassette tapes with it. I tried reversing inputs and even tried two different recorders, varying tone and volume positions. All components are O.K. and I get a signal at EF2. Can anybody out there offer any help? Thanks. G.R. Gilbert, 304 Vassal, Drummondville, Que. J2B 5H3.

HELP! I am trying to find a circuit for the ELF-II that will let me use dynamic memory. I have some MK4096 and TMS4027 dynamic memory chips but can't figure out how to arrange the refresh to avoid disturbing the timing of the 1861 video. Does anyone have dynamic memory in use with an 1802? or does anyone have any suggestions? Also, are there enough US members of ACE to have some sort of East Coast meeting? Joe Rothberg, 2908 New Castle Ave., Silver Spring, MD, 20910 USA Ph. (301)585-7322

If there are any other ACE members in this area (northern Ohio) I would enjoy hearing from them. Dick DeLombard, 802 Seneca Ave., Huron, Ohio, 44839, USA.

I'm looking for some people who might be interested in sharing the work of developing a Tiny Pilot based on the flowcharts by Allen S. Krieger in the April 78 issue of KILOBAUD? Sincerely yours, David A. Hersker, Box 802, Conyngham, Pa 18219, USA.

On my Netronics Elf, the video display has a little sideways shimmy that waves down the screen about once a second. An oscilloscope shows clean sync pulses with no 60 cycle hum and only 0.3 mV of 2Mc clock noise. Seemingly every video/sync pulse combination has been tried to no avail. Direct video input to monitors gives the same little wobble as when an RF modulator is used to a TV. If anyone else has met and conquered this problem I would love to know how. Kenneth A. Mantei, 1254 E. 26th Street, San Bernardino, California, 92404. (Ed. Note: See G. Fournier's article in this issue.)

I would like to know if you have any information on a larger BASIC or FORTRAN for the 1802. I currently have TINY BASIC up and running, but would like FLOATING POINT and a faster execution time. Thank you. Bill Freymuth, 9119 E. 66 St., Tulsa, Oklahoma, USA 74133.

A question: have any of the members worked up a SIMP-A (Standard Interface for Micro Peripherals as described by the Central Standards Library, 128 Faft Rd., Denver, Colo.)interface for an 1802 system, the VIP in particular? Thanks for your co-operation. Sincerely, Steve Place, 72 Burgundy Hill Lane, Middletown, CT, 06457.

I am writing on behalf of a small 4 people 1802 users group which includes a Complete Netronics Elf II system, a partial Elf II system, a complete Homebrew Elf system, and my own "Complete" Quest Super Elf System (only 4K at present but with S100 Interface).

I am looking for a good (better than Netronics) Character Generation and Video interface system. Would also like a Full featured 8K or more Basic for 1802. As Tiny Basic is firm but not enough. Will consider Cassette exchanges. Thank you. Kenneth J. Lynch, 47-05 48th St., Woodside, N.Y., 11377.

I would be interested in hearing from any members in my area. I presently own an Elf I but also have access to a KIM I, 8085 kit, and soon a Starplex developement system by NCS. Best Wishes. Jan Pottinger, 505 E. Lakeside Dr., Florence, Ala., 35630, USA.

STUDIO II CONVERSION KIT

FOR IMMEDIATE RELEASE DECEMBER 26, 1978

Norristown, PA - - ARESCO has announced the availability of information and components to convert the RCA Studio II video game into a simple microcomputer. The information package describes how to construct a new cartridge for the Studio II, includes schematics, ROM monitor listing, operating instructions, and program listings, and is available for $5.00. ARESCO also offers a printed circuit board and pre-programmed PROM containing the conversion program, as well as a fully assembled and tested cartridge. No modifications to the Studio II are required. With the new cartridge, the user can enter machine language programs through the Studio II keypads, and the programs can be controlled from the keypads and generate TV displays on a standard TV attached to the Studio II. The converted Studio II is a simple, powerful microcomputer containing 512 bytes of RAM, 256 bytes of ROM, an RCA COSMAC 1802 microprocessor, a video graphics interface, an FCC-approved RF modulator, two ten-key keypads, and a processor controlled "beeper".

Orders for the information package, which contains pricing for the components as well as for the assembled cartridge, should be sent to ARESCO, P.O. Box 43, Audubon PA 19407. MC/VISA/BAC orders accepted, but no C.O.D. or billing available. For more information or rush orders, contact Rick Simpson by telephone at (215) 631-9052 or (215) 631-9257.

TIC TAC TOE WITH THE 1802

Dave Burniston
203-3576 Peter St.
Windsor, Ont. N9C 1J5

I would like to make a few comments on what I have been doing with my TEC-1802 in the recent weeks. When I purchased my kit,I also bought an 1861 videa chip, which I have on a second board, along with a cassette interface and a 1.78 MHz clock.

When the November issue of Popular Electronics came out with Ed McCormick's light pen input TIC TAC TOE program, I decided it was time for some more memory. So I dug up the two 2114 memory chips that I had and went to work. Using the circuit in Fig. 1,I wire wrapped a board which allows the original 256 bytes to be addressed from 0000 - 00FF and the new 1K bytes to be addressed from 0400 - 07FF. These locations were used to comply with the TIC TAC TOE program. By replacing the original 256 bytes with two more 2114's, a complete 2K can be run.

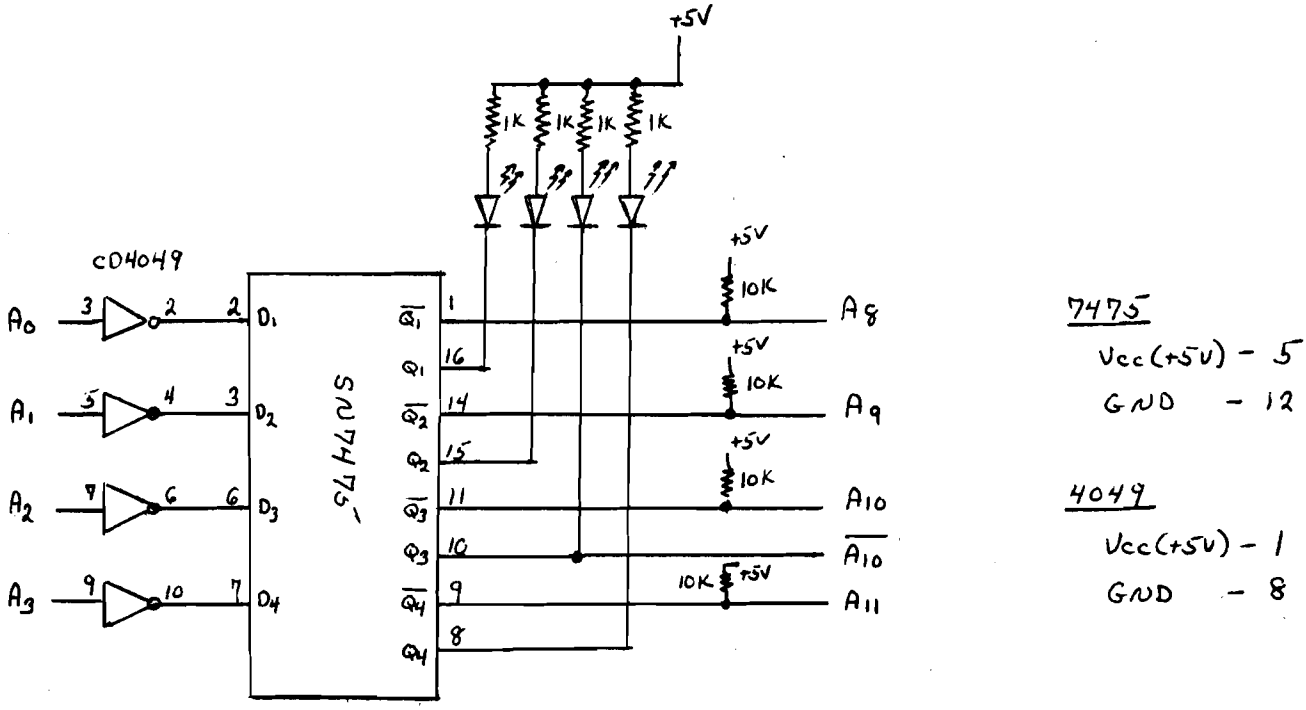


FIG. 1

The two signals A10 and $\overline{A10}$ are used as $\overline{CE}$ signals. A10 goes to the $\overline{CE}$ of the original 256 bytes, and $\overline{A10}$ goes to the $\overline{CE}$ of the newly added 2114's. Address bits A0 - A9 also go to the 2114's as shown in Fig. 2.

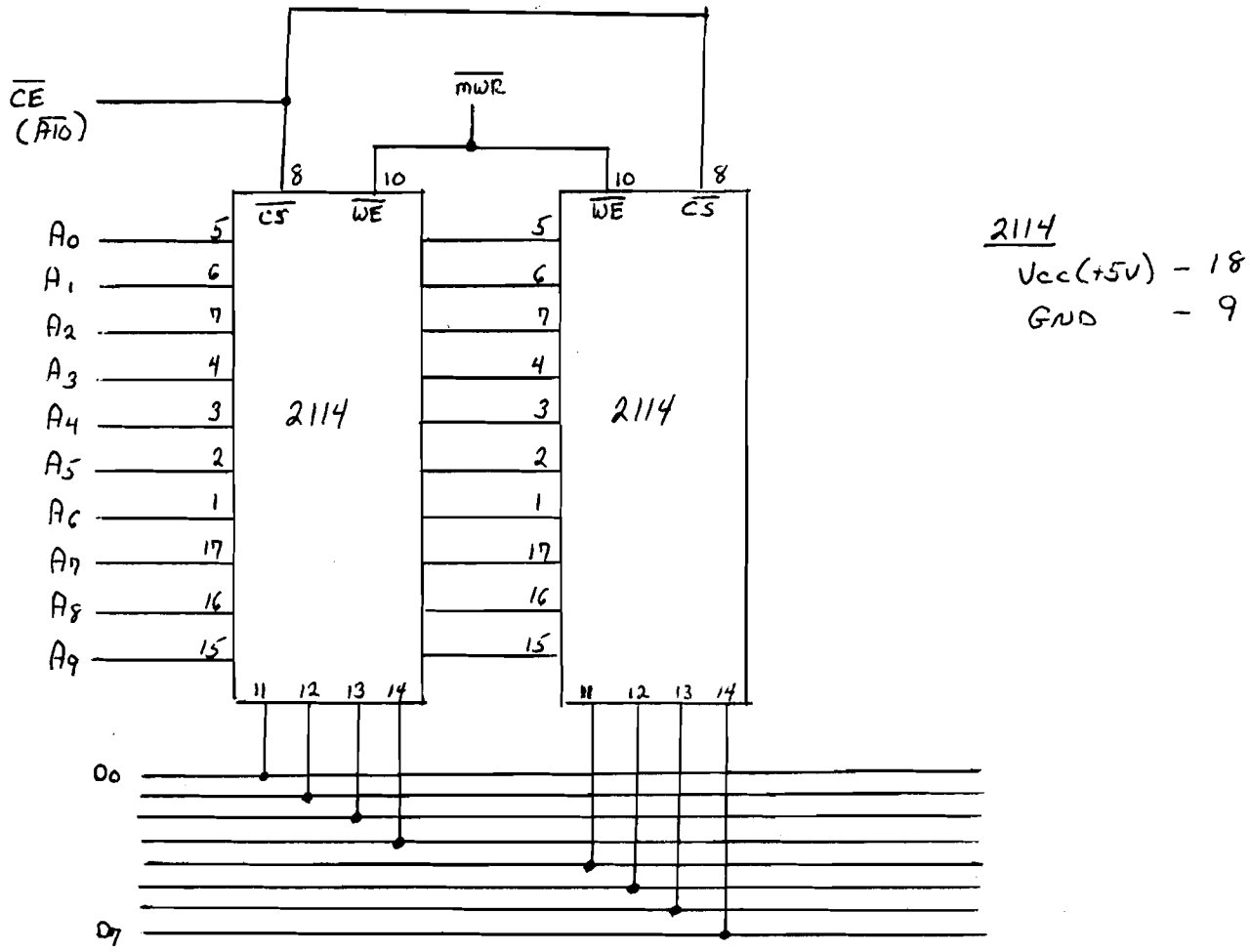


FIG. 2

Having completed this circuit, I loaded the Elf II Expansion Monitor into the original 256 bytes and used it to load the TIC TAC TOE program. Using the suggested light pen, a question mark kept being displayed when the players move was made, and no X would be displayed. My logic probe indicated that the photoresistor was constantly holding the $\overline{EF3}$ line in the low state. This problem was solved by putting a variable resistor in parallel with the $\overline{EF3}$ pull-up resistor and adjusting it until the logic probe just indicated a not-low state with the light pen on a dark square.

The program now runs fine on my system and when the computer makes the first move in a game, in most cases it is difficult to beat the computer. Even a draw is sometimes difficult, with the computer winning a surprising number of games. So be on your toes when playing this game.

Now some information on my video monitor. I am using a Carillon 110/12 V, AC/DC black and white television which I purchased from a Shoprite Catalogue Store. A schematic is supplied with the set, making conversion rather easy. I added the circuit in Fig. 3 to convert the set.

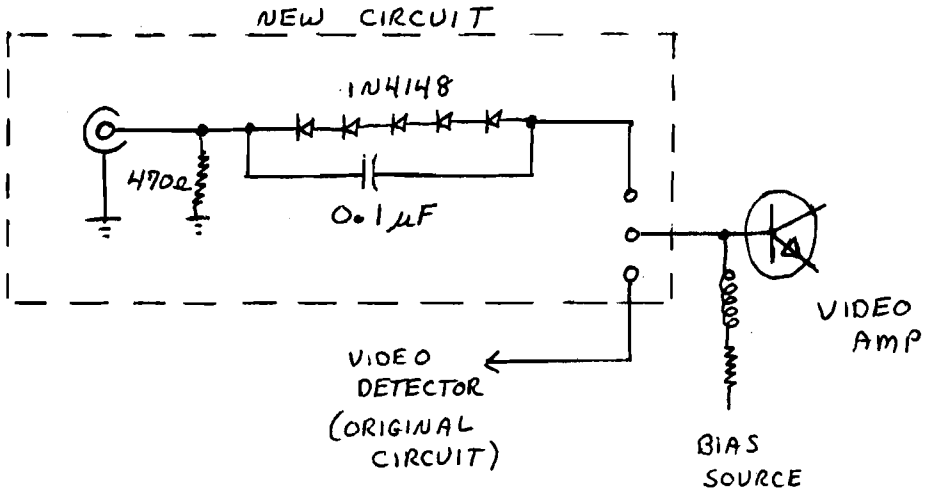


FIG. 3

The added circuit provides biasing for the video amp at the "blacker than black" level so that the TV screen is dark when there is no video signal. Adding a 1-2 volt peak to peak video signal brings the bias back up to the white level. The switch allows the set to be used either as a monitor or as a TV.

References

1. Popular Electronics, November, 1978

2. Popular Electronics, March, 1977

3. Popular Electronics, March, 1978

4. TV Typewriter Cookbook, D. Lancaster, Sam's Publications

KEY-IN LOADER FOR ELFII FORMAT TAPES

Tom Jones
1751 King Ave.
Florence, S.C.
29501

If you have a homebrewed, or another brand of 1802CD system, this utility program will allow you to read tapes formatted on a Netronics LTD ElfII without having their entire monitor in your system. They also sell some nice software packages in this format.

Insert the first location of memory you wish to start loading to in locations 0002 and 0005. Insert the range in locations 0009 and 000C. Range is the block length of the record on tape in hex, and is the same as the highest location loaded to if loading to starting locations at page boundry. Otherwise you must calculate it by finding the difference between highest and lowest addresses.

The loader gives no indication until the program is loaded unless a parity error occurs, where upon the program halts and "EE" is displayed. If the load is successful "05" is displayed. ("05" looks like O/S to me and I normally return to the operating system when a job is done.)In the event that you don't know the length of the program you are loading (bad news) you will get an "EE" if you specify too large a range and an "05" if too low a range but you will get short-changed on bytes. By bracketing the range and then increasing it until you get the first parity, you may be able to determine the unknown record length.

There are two timing constants in the loader, presently set up for a system with a 1.79 MHZ clock. The following formulas will calculate them for systems using other clocks. At location 0012, constant "F9" sets the minimum number of marks needed to recognize a valid leader. This turns out to be around .7 sec. of a 10- 12 sec. leader, interesting enough. At location 0041 is the "bit delay" constant.

1. First calculate "bit delay".

$$C_1 = \frac{2,908,380}{M}$$

Where: $M = \frac{\text{Your clock Freq (HZ)}}{8}$

C_1 = new constant (in decimal, convert to Hex)

2. Use C_1 above to calculate "X", and then C_2. Insert C_1 at location 0041 and C_2 at location 0012.

$$X(1536 \times C_1 + 2048) = (.689 \times M) - 42$$

$$C_2 = 256 - X \quad \text{(in decimal, convert to Hex)}$$

Bill Freymuth's VIP to ELF conversion utility should allow reading VIP generated tapes and converting them, also, but I have not done it yet.

Ref. Ipso Facto #8, p. 21, "VIP to ELF/ELF to VIP program"

```
LOCN OBJ CODE   STMT     SOURCE STATEMENT                    1802 VER 1.6

                   1 ..        NETRONICS FORMAT BOOTSTRAP LOADER
0000       =0100   2 ADDR      EQU   #0100          ..LOAD ADDRESS
0000       =00FF   3 RANGE     EQU   #00FF          ..LOAD RANGE
0000 E2            4 CASBOOT:  SEX   R2             ..SET UP X REG
0001 F801          5           LDI   A.1(ADDR)      ..ADDRESS MSB
0003 BA            6           PHI   RA             ..SAVE AWAY
0004 F800          7           LDI   #00            ..ADDRESS LSB
0006 AA            8           PLO   RA
0007 B7            9           PHI   R7
0008 F800         10           LDI   A.1(RANGE)     ..RANGE MSB
000A BC           11           PHI   RC
000B F8FF         12           LDI   A.0(RANGE)     ..RANGE LSB
000D AC           13           PLO   RC
000E F840         14           LDI   A.0(BITREAD)
0010 A7           15           PLO   R7
0011 F8F9         16 SYNC1:    LDI   #F9            ..SET UP CONSTANT
0013 BD           17           PHI   RD
0014 D7           18 SYNC2:    SEP   R7             ..CALL BITREAD
0015 3B11         19           BNF   SYNC1
0017 9D           20           GHI   RD
0018 3A14         21           BNZ   SYNC2          ..RESYNC
001A D7           22 INSYNC:   SEP   R7             ..LOOK FOR
001B 331A         23           BDF   INSYNC         ..START BIT.
001D F801         24           LDI   #01            ..SET UP
001F BD           25           PHI   RD             ..BIT AND
0020 AD           26           PLO   RD             ..PARITY COUNTERS.
0021 D7           27 READ1:    SEP   R7             ..CHECK
0022 9D           28           GHI   RD             .. FOR
0023 7E           29           SHLC                 ..  8
0024 BD           30           PHI   RD             ..   BITS
0025 3B21         31           BNF   READ1          ..    READ.
0027 D7           32           SEP   R7             ..OK, GET PARITY
0028 8D           33           GLO   RD             ..CHECK PARITY
0029 F6           34           SHR                  ..BIT
002A 334E         35           BDF   PARITY         ..PUNT IF ERROR
002C 9D           36           GHI   RD
002D 5A           37           STR   RA
002E 8A           38           GLO   RA
002F 1A           39           INC   RA
0030 2C           40           DEC   RC
0031 8C           41           GLO   RC
0032 3A1A         42           BNZ   INSYNC
0034 9C           43           GHI   RC
0035 3239         44           BZ    DONE
0037 3D1A         45           BR    INSYNC
0039 E0           46 DONE:     SEX   R0             ..SET UP FOR IMMEDIATE MODE
003A 64           47           OUT4                 ..AND OUTPUT GOOD STUFF
003B 05           48           DC    #05            ..INDICATION.
003C E2           49           SEX   R2             ..SET UP X AND
003D 00           50           IDL                  ..WAIT FOR SOMETHING TO HAPPEN
003E 1D           51 BITRET1:  INC   RD
003F D0           52 BITRET0:  SEP   R0             ..RETURN TO CALLER
0040 F80D         53 BITREAD:  LDI   #0D            ..SET UP 'CONSTANT'
0042 3542         54           B2    *              ..WAIT FOR HIGH-LOW TRANSITION
0044 353E         55 BIT1:     B2    BITRET1        ..IF EF2 GOES HIGH BEFORE
0046 FF01         56           SMI   #01            ..TIMEOUT, BIT IS A ONE; OTHER
0048 3344         57           BPZ   BIT1           ..WAIT FOR
004A 3D4A         58           BN2   *              ..LOW-HIGH TRANSITION.
LOCN OBJ CODE   STMT     SOURCE STATEMENT                    1802 VER 1.6

004C 303F         59           BR    BITRET0
                  60 ..       COME HERE IF PARITY ERROR
004E E0           61 PARITY:   SEX   R0             ..SET UP FOR IMMEDIATE OUTPUT
004F 64           62           OUT4                 ..DISPLAY
0050 EE           63           DC    #EE            ..  EE AS
0051 E2           64           SEX   R2             ..   THE ERROR CODE.
0052 00           65           IDL                  ..WAIT FOR SOMETHING TO HAPPEN
0053              66           END
   0 DIAGNOSTICS GENERATED
  13 SYMBOLS
SYMBOL TABLE:
ADDR      0100    RANGE     00FF    CASBOOT   0000    SYNC1     0011
SYNC2     0014    INSYNC    001A    READ1     0021    DONE      0039
BITRET1   003E    BITRET0   003F    BITREAD   0040    BIT1      0044
PARITY    004E
```

LIFE for an ELF

David A. Hersker
439 S. River St., Apt #5
Wilkes-Barre, PA 18702
USA

After reading Ben Hutchinson, Jr.'s "Game of Life" in Issue 6 of IPSO FACTO I wanted to get it up and running on my Elf II. After several false starts I succeeded, using the "Giant Board" system monitor and "Elf-Bug" (both from Netronics). Then after correcting two loading errors (thanks to suggestions from Ben and a hint in Harley Shanko's letter in Issue 8) I was off and running.

Unfortunatly, using "Elf-Bug"as an operating system used an extra 4+ pages of memory (LIFE runs in only 4 pages!) so I sat down to write an operating system tailored to LIFE and the ELF.

What follows is my second version. This system assumes that page 2 is your initial display page but can be changed to operate with any other page. It allows you to input data to any memory location on the page except FF. Keying in FF for an address returns control to the main program.

The only change you need to make in the original program (the change is optional, but it does simplify operation) is to change the Long Branch at address 0027_H from C0 01 60 to C0 01 B5. This eliminates the need to change that C0 to C8 to run the program. In addition it lets you turn the run switch off and still pick up where the program left off by holding the Input key down when you turn Run back on. The code is written in two blocks only because I didn't want to relocate the "optional" display routine since I do use it.

To run the GAME OF LIFE after loading the main program and this operating system simply turn the run switch on. At that point the Q led should come on (if not you've got a bug in the works). Q is used as an input prompt. When Q is on you input a low address byte by keying in the address and pushing input. Q will then go off and you input the data you want at that location. (Q on = put in address, Q off = put in data) Continue doing that until you've loaded all your data, then, with Q on key in FF and push the Input key. That will start the main program running.

LOCN	OBJ CODE	COMMENT
0027	C0 01 B5	This change in the original program links the main program to the operating system so you don't have to keep changing the C0 to C8.
016D	F8 01 B8	Register 8 is used as an input storage pointer.
70	F8 C0 A8	The C0 can be changed to any free 1 byte location.
73	E8	
74	F8 02 B9	Initial display page pointer. (Change if not using page 2.)
0177	7B	Q on - ready for address input.
78	3F 78	Wait for Input key to be pressed
7A	37 7A	and released.
7C	30 A0	Branch to second block of code.

LIFE for an ELF (Cont'd)

```
LOCN  OBJ CODE     COMMENTS

01A0  6C           Get keypad data input.
  A1  64           Display data on led's.   } Low Address.
  A2  28           Keep register 8 at 01C0.
  A3  A9           Sets register 9 to location to be changed.
  A4  FC 01        Ready to run program? (Address FF was last input.)
01A6  32 B3        Yes-goto 01B3; No-continue.
  A8  7A           Q off - ready for data input.
  A9  3F A9        Wait for Input key to be pressed
  AB  37 AB         and released.
  AD  6C 64 28     Input data.
01B0  59           Store data via register 9.
  B1  30 77        Continue program (loop).
  B3  7A           Q off - ready to run main program.
  B4  C8           Skip next 2 bytes and prepare to run program.
  B5  3F 60        If Input key up goto Erase subroutine and then to
                    operating system to input data.  If Input key
                    down run program.
01B7  C0 00 2D     Branch to main program and run.
```

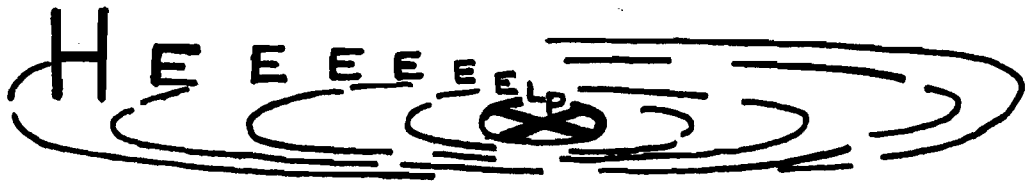
If you'd like some more background (and patterns) on the GAME OF LIFE in addition to the SCIENTIFIC AMERICAN articles cited by Ben try the December, 1978 issue of BYTE. There are several articles about LIFE.

The "Glider Gun" in the first article is the same as the first gun in the second article. They're just at different stages of growth. Happy LIFEing.

BACK ISSUE REPRINT POLICY

The cost of reprints of back issues will be:

1 complete set of issues #1 to 6 ..... $12.00
Per copy price of single issue ..... $ 2.50



Tom Crawford
Past-President
ACE

The Association of Computer Experimenters is well into its second year of existence. It is time to consider the future of ACE, specifically that period from June 1979 to May 1980.

Our current executive members, who have been working in this capacity since June 1978, have done an excellent job of maintaining and building the club. The newsletter has grown in both content and quality. The monthly club meetings are even more interesting thanks to the presentations by local electronics and computer-related business representatives. And the bi-weekly tutorials continue to meet with a great deal of interest for both the hardware and software material presented. In addition, the work load involved in running the club and putting out the newsletter has been spread over a larger number of people this year. This results in both less work for each individual, and a larger amount of effort expended overall, resulting in the improvements mentioned above.

The dedication and abilities of these people, who number about 10, has resulted in amazing success for the club and newsletter. However, these people have served their turn, and should be replaced in the upcoming club elections.

The problem is to maintain a high quality, dynamic club and still not overwork individuals. The solution is the careful choice of a few key executive and a lot of volunteer working committee people.

The particular area that requires a number of "volunteer" workers is the newsletter. Maintaining a current mailing list of over 450 names and addresses can be a formidable task. The problems of getting the newsletter printed, collated, stuffed into envelopes and mailed can keep 4 people occupied. Editing the newsletter is a task in itself. Another couple of volunteers to take care of typing and diagrams would be appreciated.

The direction the club will take under a new President has yet to be determined. However, some form of software, hardware, standards, membership and program co-ordination has to be maintained. In most of these cases the tasks are now handled by a committee of one. I suggest a committee of at least 2 would lighten the load and tend to keep the ideas and objectives progressing.

The purpose of this article is, therefore, to point out some of the facts of club life, and to issue a call for nominations or volunteers.

NOMINATIONS AND ELECTIONS

The procedures involved are specified in the ACE Constitution (on pgs 39-41 of Issue #3 of IPSO FACTO, or contact the Secretary/Treasurer, George York). Elections are held at the Annual General Meeting (Tuesday, May 8, 1979), and the following election procedures apply:

NOMINATIONS AND ELECTIONS (CONT'D)

1. A nominating committee shall, as soon as possible, consist of 3 past presidents and be chaired by the most immediate past president. For the present, a nominating committee selected and chaired by the past president, shall propose a slate of new executive.
2. The proposed slate shall be published at least six weeks prior to the election, with a proviso that further nominations must reach the nominating committee chairman in writing (note: nominating committee chairman--Tom Crawford, 50 Brentwood Dr., Stoney Creek, Ont. L8G 2W8) at least one month prior to the election.
3. The proposed slate shall consist of at least one nominee for the positions of president, secretary, treasurer, and newsletter editor plus a number of committee executive members.
4. Election and/or ratification of the proposed slate shall be carried out at the annual meeting. If there are no contested positions on the slate, ratification of the slate shall be by a show of hands of those members present. If there are contested positions, election shall be by secret ballot as per the Constitution.

THE ASSOCIATION OF COMPUTER EXPERIMENTERS MINUTES OF CLUB MEETING 78-9 HELD AT STELCO WILCOX ST. AUDITORIUM 14 NOVEMBER, 1978 8:00 P.M.

78-9-1 The meeting was preceded by a 1 hour tutorial.

78-9-2 Wayne Bowdish reported a paid membership of 310.

78-9-3 Rod Dore reported the tutorials will be looking at cassette interfaces. The tutorials will start at 7:30 P.M. on nights when only tutorials are held.

78-9-4 The next newsletter will be out shortly after the New Year.

78-9-5 Motion to adopt Minutes 78-7 and 78-8 as included in Newsletter Issue #8.
Proposed - George York
Seconded - Bob Simpson
Carried unanimously.

78-9-6 There was no financial report.

78-9-7 Bert deKat discussed a number of devices eg. humidity detectors, PH electrodes, reed and mercury switches, Display drivers and how they could be supplied with microprocessors.

78-9-8 Don McAdam, General Manager of Varah's gave a very interesting talk about memory systems.

78-9-9 Motion to adjourn meeting
Proposed - George York
Seconded - John Hanson
Passed
The meeting adjourned at 10:30 P.M. About 26 people attended.

THE ASSOCIATION OF COMPUTER EXPERIMENTERS MINUTES OF CLUB MEETING 78-10 HELD AT STELCO WILCOX ST. AUDITORIUM 12 DECEMBER, 1978 8:00 P.M.

78-10-3 The meeting was preceded by a tutorial.

78-10-2 George York reported a current bank balance of $2411.66.
Wayne Bowdish reported a paid membership of 329.
Rod Dore reported tutorials are progressing with cassette interfaces as the main topic.
Bernie Murphy mentioned the shortage of locally generated newsletter articles.

78-10-3 Eugene Tekatch gave a talk on developments in 1802 systems and technology.

78-10-4 There was a demonstration of a VIP and a Tektron system.

78-10-5 There were a number of entries for the T-shirt Logo Contest. Members at the meeting voted for their choice. The winning entry will be published in the newsletter.

78-10-6 The meeting adjourned at 11:00 P.M.
About 32 people attended the meeting.

THE ASSOCIATION OF COMPUTER EXPERIMENTERS MINUTES OF CLUB MEETING 79-1 HELD AT STELCO WILCOX ST. AUDITORIUM 9 JANUARY, 1979 8:00 P.M.

79-1-1 The meeting was preceded by a 1 hour tutorial.

79-1-2 George York, secretary/treasurer, reported a current membership of 403 paid members and a current bank balance of $3318.53.

79-1-3 Motion to adopt Minutes of 78-9 and 78-10.
Proposed - George York
Seconded - John Norris
Unanimous

79-1-4 Rod Dore is developing some suitable hardware for cassette interface and tutorial demonstration.
Fred Feaver mentioned a bug in the Tektron system when addressing. The fix will be published in the newsletter.
Michael Franklin mentioned he had a problem with the Netronics Elf.

79-1-5 Dave Walton, Manager of Radio Shack (Dundas), gave a talk on TRS-80 systems. Three units were available for demonstrations.

79-1-6 The meeting adjourned at 10:45 P.M.
26 people attended the meeting.

IEEE MICROPROCESSOR COURSE NOTICE

The IEEE Hamilton Section will be running a Spring, 1979, course called "Microprocessor Programming and Interfacing Techniques (RCA-1802)". The previous "Microprocessors for Industry" course including the TEC-1802 microprocessor kit, will be a prerequisite. The new course will provide:

- review of 1802 programming
- an introduction to Interpretive programming and Macro instructions
- Editor, Monitor, BASIC
- Cassette, Teletype and Video interfacing
- EPROM Programming
- Industrial application techniques.

Included in the course fee is the additional .75K RAM board required to run the larger programs needed during this course.

Courses will all be given at Chedoke Hospitals' Nash Auditorium. To register please indicate courses in order of preference (1, 2, 3 etc.). Also indicate any optional hardware you wish to purchase. Mail the registration with a cheque payable to IEEE Hamilton Section for the total payment to: IEEE Training Committee, 196 Homewood Avenue, Hamilton, Ontario, L8P 2M5.

IEEE MEMBER FEES APPEAR IN ( ) PLEASE CHECK ( )

1. Refresher course (bring notes "Microprocessors for Industry" and TEC-1802 microprocessor)
   Saturday, February 3, 1979 8:30 a.m. to 5:00 p.m. with break for lunch. $45.00 (35.00) ( )

Advanced Courses (memory expansion kit incl.) $150.00(130.00)

2. February 7 to March 28, 1979 - Wed. evenings 7-10p.m. ..... ( )
3. April 4 to May 23, 1979 - Wed. evenings 7-10p.m. ..... ( )
4. April 21, April 28, and May 12, 1979 ..... ( )
   3 Saturdays 8:30 a.m. to 5:00 p.m. (break for lunch)

Relevant Optional Hardware for Advanced 1802 Course

TEC-1802-MB2/7K RAM with space for 2K EPROM	$210.00	..... ( )
Above with 3K of RAM only	$135.00	..... ( )
TEC-1802-IF1 interface board with cassette, teletype and video interfaces	$ 55.00	..... ( )

NAME _____ IEEE MEMBERSHIP NUMBER _____

ADDRESS _____

_____

PHONE (Home) _____ (Business) _____

PLEASE RETURN FORM AND PAYMENT AS SOON AS POSSIBLE

Unless you receive further notification assume that you are enrolled in your first choice course.

NOTICE OF CLUB MEMBERSHIP RENEWAL & NEW CLUB DUES ASSESSMENT

All club memberships expire 31 May, 1978. Club members should complete the form below and mail to George York, 60 Chester Road, Stoney Creek, Ontario, Canada, L8E 1Y2.

The club dues assessment for the 1978-79 fiscal year is $10.00. Make cheques payable to the Association of Computer Experimenters.

If you know of anyone who would be interested in joining our club, why not give him/her the membership application. We are constantly looking for new members with new and interesting ideas.

MEMBERSHIP APPLICATION FORM FOR THE Association of Computer Experimenter

FIRST NAME | LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE | POSTAL CODE

Association of Computer Experimenters

MEMBERSHIP RENEWAL FORM

ATTACH MAILING LABEL HERE

FIRST NAME | LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE | POSTAL CODE